ALIBABA CLOUD

Alibaba Cloud

Apsara Stack Enterprise

User Guide - Cloud Essentials and Security

Product Version: V3.13.0

Document Version: 20220526

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Apsara Uni-manager Management Console

1.1. User Guide

1.1.1. What is the Apsara Uni-manager Management Console?

The Apsara Uni-manager Management Console is a service capability platform based on the Alibaba Cloud Apsara Stack platform and designed for government and enterprise customers. This platform improves IT management and troubleshooting and is dedicated to providing a leading service capability platform of the cloud computing industry. It provides large-scale and cost-efficient end-to-end cloud computing and big data services for customers in industries such as government, education, healthcare, finance, and enterprise.

Overview

The Apsara Uni-manager Management Console simplifies the management and deployment of physical and virtual resources by building an Apsara Stack platform that supports various business types of government and enterprise customers. The console helps you build your business systems in a simple and quick manner, fully improve resource utilization, and reduce O&M costs. This allows you to shift your focus from O&M to business. The console brings the Internet economy model to government and enterprise customers, and builds a new ecosystem chain based on cloud computing.

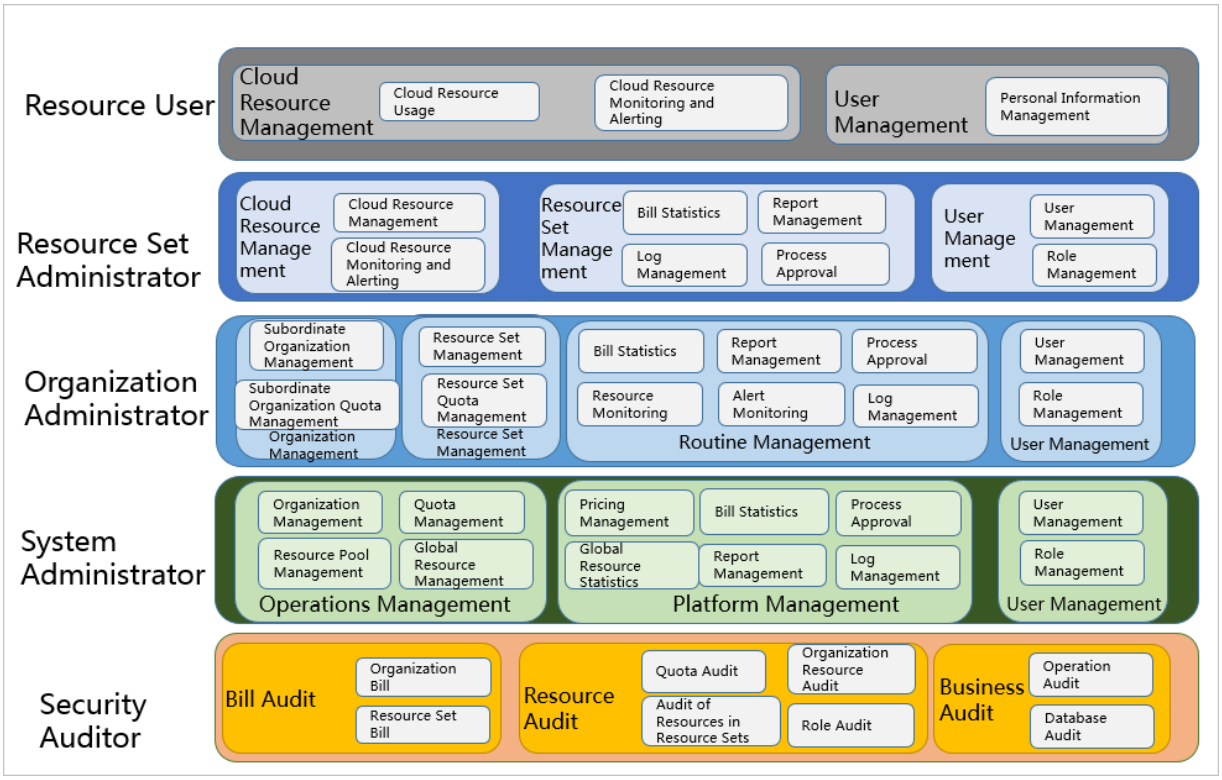
Workflow

Operations in the Apsara Uni-manager Management Console are divided into the following parts:

- System initialization: This part is designed to complete basic system configurations, such as creating organizations, resource sets, and users, creating basic resources such as VPCs, and creating contacts and contact groups in Cloud Monitor.
- Cloud resource creation: This part is designed to create resources.
- Cloud resource management: This part is designed to complete resource management operations, such as starting, using, and releasing resources, and changing resource configurations and resource quotas.

1.1.2. User roles and permissions

This topic describes roles and their permissions.



Roles and permissions

Role	Role permission
Resource user	This role has the permissions to view and modify resources in a resource set and create alert rules.
Resource set administrator	This role has the permissions to create, modify, and delete resources in a resource set and manage the users of the resource set.
Organization administrator	This role has the permissions to manage an organization and its subordinate organizations, create, modify, and delete the resources of organizations, create and view alert rules for resources, and export reports.
Operations administrator	This role has read and write permissions on all resources.
Security auditor	This role performs security audit on the Apsara Uni-manager Management Console and has the read-only permissions on operation logs of the Apsara Uni-manager Management Console.
Platform administrator	This role has the permissions to initialize the system and create operations administrators.
Resource auditor	This role has the read-only permissions on all resources in the Apsara Uni-manager Management Console.
Organization security administrator	This role manages the security of an organization, including the security of hosts, applications, and networks. This role has the read-only permissions on operation logs of the Apsara Uni-manager Management Console and read and write permissions on ApsaraDB RDS, ECS, and Apsara Stack Security.

Role	Role permission
Security system configuration administrator	This role configures system security features such as the upgrade center and global configurations. This role has read and write permissions on the upgrade, protection, and configuration features of Apsara Stack Security.
Global organization security administrator	This role manages the security of global tenants by using Cloud Security Operation Center (SOC). This role has read and write permissions on all features of Apsara Stack Security.
Platform security administrator	This role manages the security of the Apsara Uni-manager Management Console by using SOC.
Global organization security auditor	This role checks the security conditions of all organizations by using SOC. This role has the read-only permissions on operation logs of the Apsara Uni-manager Management Console and all features of Apsara Stack Security.
Platform security auditor	This role checks the security conditions of the Apsara Uni-manager Management Console by using SOC. This role has the read-only permissions on operation logs of the Apsara Uni-manager Management Console, Server Guard, Cloud Firewall, Sensitive Data Discovery and Protection, SOC, system configurations, and Web Application Firewall (WAF) configurations as well as read and write permissions on Anti-DDoS, Threat Detection, and Update Center of Apsara Stack Security.
Platform security configuration administrator	This role configures and has read and write permissions on security services in the Apsara Uni-manager Management Console, such as Server Guard and WAF.
Organization resource auditor	This role has the read-only permissions on all resources in an organization to which it belongs.

1.1.3. Log on to the Apsara Uni-manager Management Console

This topic describes how to log on to the Apsara Uni-manager Management Console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

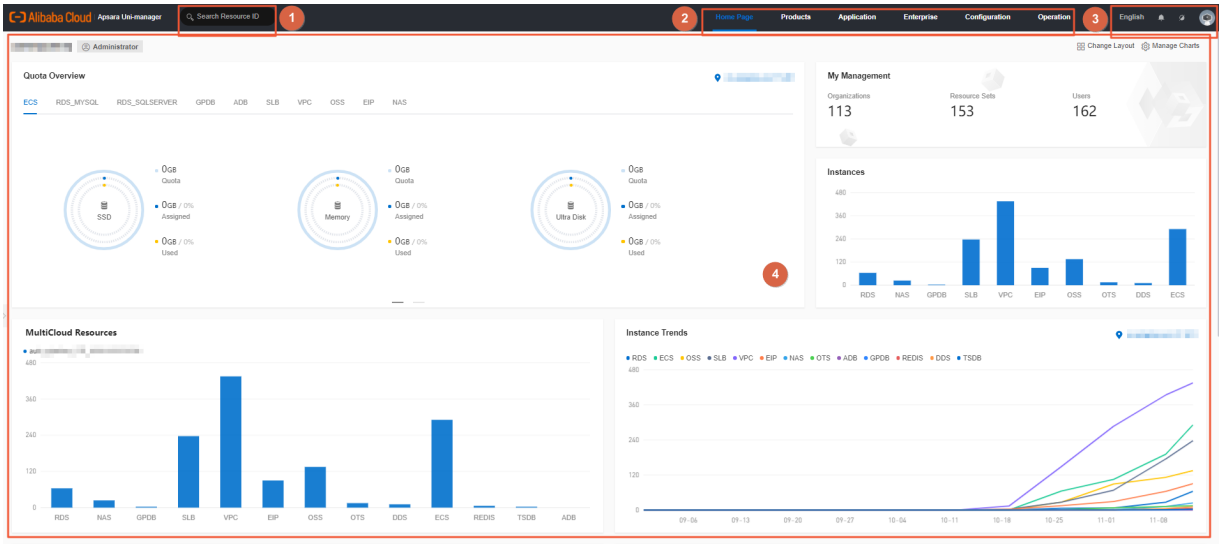
> **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
>
> - Uppercase or lowercase letters
> - Digits
> - Special characters, which include ! @ # $ %

3. Click **Login**.

1.1.4. Web page introduction

The web page of the Apsara Uni-manager Management Console consists of the search box, top navigation bar, information section of the current logon user, and operation section.

Apsara Uni-manager Management Console page



Functional sections of the web page

Section		Description
①	Search box	This section allows you to search for cloud services by resource ID.
②	Top navigation bar	This section includes the following modules: • **Home**: uses charts to display the usage and monitoring data of system resources in each region. • **Products**: manages all types of basic cloud services and resources. • **Enterprise**: manages organizations, resource sets, roles, users, logon policies, user groups, ownership, and resource pools. • **Configurations**: manages resource pools, password policies, specifications, menus, and RAM roles. • **Operations**: manages the daily operations of cloud resources, including usage statistics and quotas. • **Security**: provides operations logs and system logs.

Section		Description
③	Information section of the current logon user	• English: allows you to switch between English, simplified Chinese, and traditional Chinese. • : allows you to switch between day and night modes. • User Information: When you click the icon of the current logon user, the **User Information**, **View Version**, and **Exit** menu items are displayed. ◦ If you click **User Information**, you can perform the following operations on the User Information page: ▪ View basic information. ▪ Modify personal information. ▪ Change the logon password. ▪ View the AccessKey pair of your Apsara Stack tenant account. ▪ Switch the current role. ▪ Enable or disable alert notification. ◦ If you click **View Version**, you can view the version, authorization status, and build number of Apsara Stack in the message that appears. ◦ If you click **Exit**, you can log off from the current account.
④	Operation section	**Operation section**: shows the information and operations.

1.1.5. Initial configuration

1.1.5.1. Configuration description

Before you use the Apsara Uni-manager Management Console, you must complete a series of basic configuration operations as an administrator, such as creating organizations, resource sets, users, and roles and initializing resources. This is the initial system configuration.

The Apsara Uni-manager Management Console manages the organizations, resource sets, users, and roles of cloud data centers in a centralized and service-oriented manner to grant different resource access permissions to different users.

- Organization

  After the Apsara Uni-manager Management Console is deployed, a root organization is automatically generated. You can create other organizations under the root organization.

  Organizations are displayed in a hierarchical structure. You can create subordinate organizations under each organization level.

- Resource Set

  A resource set is a container used to store resources. Each resource must belong to a resource set.

- User

  A user is a resource manager and user.

- Role

A role is a set of access permissions. You can assign different roles to different users to implement system access control to meet a variety of different requirements.

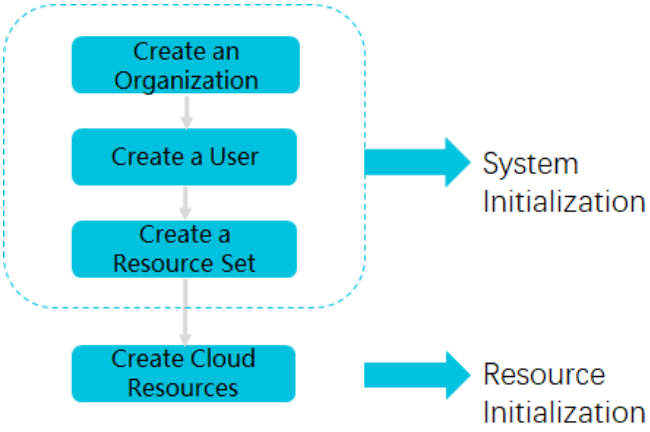
The following table describes the relationships among organizations, resource sets, users, roles, and cloud resources.

Relationship between two items	Relationship type	Description
Organization and resource set	One-to-many	An organization can have multiple resource sets, but each resource set can belong to only a single organization.
Organization and user	One-to-many	An organization can have multiple users, but each user can belong to only a single organization.
Resource set and user	Many-to-many	A user can have multiple resource sets, and a resource set can be assigned to multiple users under the same level-1 organization.
User and role	Many-to-many	A user can have multiple roles, and a role can be assigned to multiple users.
Resource set and resource	One-to-many	A resource set can have multiple resources, but each cloud resource can belong to only a single resource set.

1.1.5.2. Configuration process

This topic describes the initial configuration process.

Before you use the Apsara Uni-manager Management Console, you must complete the initial system configurations as an administrator based on the process shown in the following figure.



1. Create an organization

   Create an organization to store resource sets and their resources.

2. Create a user

   Create a user and assign the user different roles to meet different requirements for system access control.

3. Create a resource set

   Create a resource set before you apply for resources.

4. Create cloud resources

   Create instances in each service console based on project requirements. For more information about how to create cloud service instances, see the user guide of each cloud service.

1.1.6. Monitoring

1.1.6.1. View the workbench

The Apsara Uni-manager Management Console uses charts to keep you up to date on the current usage and monitoring information of resources.

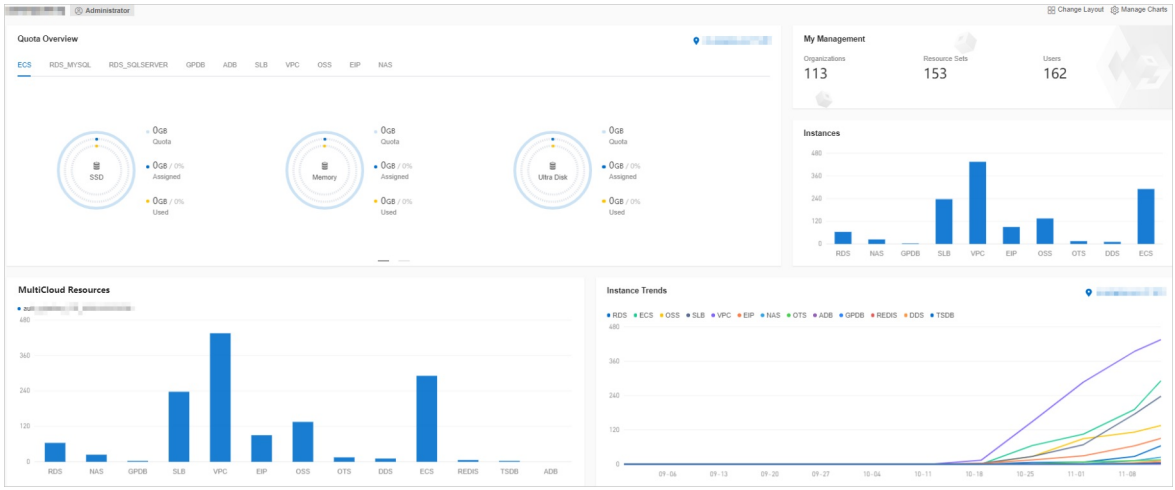
Context

> **Note** The resource types displayed may vary with region types. See your dashboard for available resource types.

Procedure

1. Log on to the Apsara Uni-manager Management Console.

   By default, the workbench page appears when you log on to the Apsara Uni-manager Management Console. To return to the workbench page from other pages, click **Home** in the top navigation bar.



2. On the workbench page, you can view the instance summary information for all regions of the Apsara Stack environment.

   You can click **Manage Charts** in the upper-right corner of the page to select all or individual modules to view relevant information. You can also click **Change Layout** in the upper-right corner of the page and drag a specific module to a location.

   - **Quota Overview**

     Shows the usage and quotas of Elastic Compute Service (ECS), ApsaraDB RDS, Object Storage Service (OSS), and Server Load Balancer (SLB) resources.

   - **Instances**

     Shows the numbers of ECS instances, ApsaraDB RDS instances, OSS buckets, and SLB instances in each region.

   - **Instance Trends**

     Shows the numbers of ECS instances, ApsaraDB RDS instances, OSS buckets, and SLB instances for the last five days.

   - **Resource Load**

     Shows the top five ECS and ApsaraDB RDS instances in terms of disk usage, CPU utilization, and memory usage.

- **Alert Rules**

  Shows the number of alerts and details of the alerts.

- **My Management**

  Shows the numbers of organizations, resource sets, and users.

- **Region Map**

  Shows the information of all primary and secondary nodes in Apsara Stack. The network connection status and related alerts are displayed for each secondary node.

- **Cloud Resource Count**

  Shows the cloud services and the number of instances in each secondary node.

1.1.6.2. CloudMonitor

1.1.6.2.1. Cloud Monitor overview

Cloud Monitor provides real-time monitoring, alerting, and notification services for resources to protect your services and businesses.

Cloud Monitor can monitor metrics for a variety of services such as ECS, ApsaraDB RDS, SLB, OSS, KVStore for Redis, VPN Gateway, AnalyticDB for PostgreSQL, ApsaraDB for MongoDB, EIP, and API Gateway.

You can use the metrics of cloud services to configure alert rules and notification policies. This way, you can stay up to date on the running status and performance of your service instances and scale resources in a timely manner when resources are insufficient.

1.1.6.2.2. Metrics

This topic describes the metrics available for each service.

Cloud Monitor checks the availability of services based on their metrics. You can configure alert rules and notification policies for these metrics to stay up to date on the running status and performance of monitored service instances.

Cloud Monitor can monitor resources of other services, including Elastic Compute Service (ECS), ApsaraDB RDS, Server Load Balancer (SLB), Object Storage Service (OSS), KVStore for Redis, VPN Gateway, AnalyticDB for PostgreSQL, ApsaraDB for MongoDB, Elastic IP Address (EIP), and API Gateway. The following tables list the metrics for each service.

Operating system metrics for ECS

Metric	Description	Unit
Host.cpu.total	The total CPU utilization of an ECS instance.	%
Host.mem.usedutilization	The memory usage of an ECS instance.	%
Host.load1	The system loads over the last 1 minute. This metric is unavailable for Windows operating systems.	N/A
Host.load5	The system loads over the last 5 minutes. This metric is unavailable for Windows operating systems.	N/A

Metric	Description	Unit
Host.load15	The system loads over the last 15 minutes. This metric is unavailable for Windows operating systems.	N/A
Host.disk.utilization	The disk usage of an ECS instance.	%
Host.disk.readbytes	The number of bytes read from the disk per second.	byte/s
Host.disk.writebytes	The number of bytes written to the disk per second.	byte/s
Host.disk.readIops	The number of read requests received by the disk per second.	count/s
Host.disk.writeIops	The number of write requests received by the disk per second.	count/s
Host.fs.inode	The inode usage.	%

Basic metrics for ECS

Metric	Description	Unit
CPU utilization	The CPU utilization of an ECS instance.	%
Inbound bandwidth to the Internet	The average rate of inbound traffic to the Internet.	bit/s
Inbound bandwidth to the internal network	The average rate of inbound traffic to the internal network.	bit/s
Outbound bandwidth from the Internet	The average rate of outbound traffic from the Internet.	bit/s
Outbound bandwidth from the internal network	The average rate of outbound bandwidth from the internal network.	bit/s
System disk BPS	The number of bytes read from and written to the system disk per second.	byte/s
System disk IOPS	The number of reads from and writes to the system disk per second.	count/s
Advance CPU credits	The changes in advance CPU credits. Advance CPU credits can be used only when the unlimited mode is enabled.	N/A
CPU credit consumption	The changes in CPU credit consumption. Consumption trends are consistent with CPU utilization.	N/A

Metric	Description	Unit
Overdrawn CPU credits	The changes in overdrawn CPU credits. Overdrawn CPU credits can be used only when the unlimited mode is enabled.	N/A
CPU credit balance	The changes in CPU credit balance. The CPU credit balance is used to maintain CPU credit usage.	N/A

> **Note**
>
> For ECS instances, you must install a monitoring plug-in to collect metric data at the operating system level.
>
> Installation method: On the **Cloud Monitor** page, select the target instance from the ECS instance list and click **Batch Install** in the lower part of the page.
>
> Metric data is displayed in the monitoring chart within 5 to 10 minutes after the monitoring plug-in is installed.

Metrics for ApsaraDB RDS for PostgreSQL

Metric	Description	Apsara Stack service	Calculation formula
CPU utilization	The CPU utilization of an ApsaraDB RDS for PostgreSQL instance. Unit: %.	ApsaraDB RDS for PostgreSQL	Used CPU cores of an ApsaraDB RDS for PostgreSQL instance/Total CPU cores of the ApsaraDB RDS for PostgreSQL instance
Memory usage	The memory usage of an ApsaraDB RDS for PostgreSQL instance. Unit: %.	ApsaraDB RDS for PostgreSQL	Used memory of an ApsaraDB RDS for PostgreSQL instance/Total memory of the ApsaraDB RDS for PostgreSQL instance
Disk usage	The disk usage of an ApsaraDB RDS for PostgreSQL instance. Unit: %.	ApsaraDB RDS for PostgreSQL	None
IOPS usage	The number of I/O requests for an ApsaraDB RDS for PostgreSQL instance per second. Unit: %.	ApsaraDB RDS for PostgreSQL	Number of I/O requests for an ApsaraDB RDS for PostgreSQL instance/Statistical period
Connection usage	The number of connections between an application and an ApsaraDB RDS for PostgreSQL instance per second. Unit: %.	ApsaraDB RDS for PostgreSQL	Number of connections between an application and an ApsaraDB RDS for PostgreSQL instance/Statistical period

Metrics for ApsaraDB RDS for MySQL

Metric	Description	Apsara Stack service	Calculation formula
CPU utilization	The CPU utilization of an ApsaraDB RDS for MySQL instance. Unit: %.	ApsaraDB RDS for MySQL	Used CPU cores of an ApsaraDB RDS for MySQL instance/Total CPU cores of the ApsaraDB RDS for MySQL instance

Metric	Description	Apsara Stack service	Calculation formula
Memory usage	The memory usage of an ApsaraDB RDS for MySQL instance. Unit: %.	ApsaraDB RDS for MySQL	Used memory of an ApsaraDB RDS for MySQL instance/Total memory of the ApsaraDB RDS for MySQL instance
Disk usage	The disk usage of an ApsaraDB RDS for MySQL instance. Unit: %.	ApsaraDB RDS for MySQL	None
IOPS usage	The number of I/O requests for an ApsaraDB RDS for MySQL instance per second. Unit: %.	ApsaraDB RDS for MySQL	Number of I/O requests for an ApsaraDB RDS for MySQL instance/Statistical period
Connection usage	The number of connections between an application and an ApsaraDB RDS for MySQL instance per second. Unit: %.	ApsaraDB RDS for MySQL	Number of connections between an application and an ApsaraDB RDS for MySQL instance/Statistical period
Inbound bandwidth to ApsaraDB RDS for MySQL	The inbound traffic to an ApsaraDB RDS for MySQL instance per second.	ApsaraDB RDS for MySQL	None
Outbound bandwidth from ApsaraDB RDS for MySQL	The outbound traffic from an ApsaraDB RDS for MySQL instance per second.	ApsaraDB RDS for MySQL	None

Metrics for ApsaraDB RDS for SQL Server

Metric	Description	Apsara Stack service	Calculation formula
CPU utilization	The CPU utilization of an ApsaraDB RDS for SQL Server instance. Unit: %.	ApsaraDB RDS for SQL Server	Used CPU cores of an ApsaraDB RDS for SQL Server instance/Total CPU cores of the ApsaraDB RDS for SQL Server instance
Memory usage	The memory usage of an ApsaraDB RDS for SQL Server instance. Unit: %.	ApsaraDB RDS for SQL Server	Used memory of an ApsaraDB RDS for SQL Server instance/Total memory of the ApsaraDB RDS for SQL Server instance
Disk usage	The disk usage of an ApsaraDB RDS for SQL Server instance. Unit: %.	ApsaraDB RDS for SQL Server	None
IOPS usage	The number of I/O requests for an ApsaraDB RDS for SQL Server instance per second. Unit: %.	ApsaraDB RDS for SQL Server	Number of I/O requests for an ApsaraDB RDS for SQL Server instance/Statistical period
Connection usage	The number of connections between an application and an ApsaraDB RDS for SQL Server instance per second. Unit: %.	ApsaraDB RDS for SQL Server	Number of connections between an application and an ApsaraDB RDS for SQL Server instance/Statistical period

Metric	Description	Apsara Stack service	Calculation formula
Inbound bandwidth to ApsaraDB RDS for SQL Server	The inbound traffic to an ApsaraDB RDS for SQL Server instance per second.	ApsaraDB RDS for SQL Server	None
Outbound bandwidth from ApsaraDB RDS for SQL Server	The outbound traffic from an ApsaraDB RDS for SQL Server instance per second.	ApsaraDB RDS for SQL Server	None

Metrics for PolarDB

Metric	Description	Apsara Stack service	Calculation formula
CPU utilization	The CPU utilization of a PolarDB instance. Unit: %.	PolarDB	Used CPU cores of a PolarDB instance/Total CPU cores of the PolarDB instance
Memory usage	The memory usage of a PolarDB instance. Unit: %.	PolarDB	Used memory of a PolarDB instance/Total memory of the PolarDB instance
Disk usage	The disk usage of a PolarDB instance. Unit: %.	PolarDB	None
IOPS usage	The number of I/O requests for a PolarDB instance per second. Unit: %.	PolarDB	Number of I/O requests for a PolarDB instance/Statistical period
Connection usage	The number of connections between an application and a PolarDB instance per second. Unit: %.	PolarDB	Number of connections between an application and a PolarDB instance/Statistical period

Metrics for SLB

Metric	Description	Unit
Inbound bandwidth on a port	The average rate of inbound traffic on a port.	bit/s
Outbound bandwidth on a port	The average rate of outbound traffic on a port.	bit/s
Number of new connections on a port	The average number of new TCP connections established between clients and SLB instances in a statistical period.	N/A
Number of inbound packets received on a port	The number of packets received by an SLB instance per second.	count/s
Number of outbound packets sent on a port	The number of packets sent by an SLB instance per second.	count/s

Metric	Description	Unit
Number of active connections on a port	The number of TCP connections in the ESTABLISHED state. If persistent connections are used, a connection can transfer multiple file requests at one time.	N/A
Number of inactive connections on a port	The number of TCP connections that are not in the ESTABLISHED state. You can run the **netstat -an** command to view the connections for both Windows and Linux instances.	N/A
Number of concurrent connections on a port	The number of established TCP connections.	count/s
Number of dropped connections on a port	The number of connections dropped per second.	count/s
Number of dropped inbound packets on a port	The number of inbound packets dropped per second.	count/s
Number of dropped outbound packets on a port	The number of outbound packets dropped per second.	count/s
Dropped inbound bandwidth on a port	The amount of inbound traffic dropped per second.	bit/s
Dropped outbound bandwidth on a port	The amount of outbound traffic dropped per second.	bit/s

Metrics for monitoring service overview of OSS

Metric	Description	Unit
Availability	The metric that describes the system availability of OSS. You can obtain the metric value based on the following formula: Metric value = `1 - Server error requests with the returned HTTP status code 5xx/All requests`.	%
Valid request percentage	The percentage of valid requests out of all requests.	%
Total number of requests	The total number of requests that are received and processed by the OSS server.	N/A
Number of valid requests	The total number of requests with HTTP status codes 2xx and 3xx returned.	N/A
Outbound traffic from the Internet	The amount of outbound traffic from the Internet.	byte

Metric	Description	Unit
Inbound traffic to the Internet	The amount of inbound traffic to the Internet.	byte
Outbound traffic from the internal network	The amount of outbound traffic from the internal network.	byte
Inbound traffic to the internal network	The amount of inbound traffic to the internal network.	byte
CDN outbound traffic	The amount of outbound traffic sent over CDN after CDN is activated. Such outbound traffic over CDN is back-to-origin traffic.	byte
CDN inbound traffic	The amount of inbound traffic received over CDN after CDN is activated.	byte
Outbound traffic of cross-region replication	The amount of outbound traffic generated during data replication after cross-region replication is enabled.	byte
Inbound traffic of cross-region replication	The amount of inbound traffic generated during data replication after cross-region replication is enabled.	byte
Storage size	The amount of total storage occupied by the buckets of a specified user before the statistics collection deadline.	byte
Number of PUT requests	The total number of PUT requests made by the user between 00:00:00 on the first day of the current month and the statistics collection deadline.	N/A
Number of GET requests	The total number of GET requests made by the user between 00:00:00 on the first day of the current month and the statistics collection deadline.	N/A

Metrics for request status details of OSS

Metric	Description	Unit
Number of requests with server-side errors	The total number of system-level error requests with the returned HTTP status code 5xx.	N/A
Percentage of requests with server-side errors	The percentage of requests with server-side errors out of all requests.	%
Number of requests with network errors	The total number of requests with the returned HTTP status code 499.	N/A

Metric	Description	Unit
Percentage of requests with network errors	The percentage of requests with network errors out of all requests.	%
Number of requests with client-side authorization errors	The total number of requests with the returned HTTP status code 403.	N/A
Percentage of requests with client-side authorization errors	The percentage of requests with authorization errors out of all requests.	%
Number of requests with client-side errors indicating resources not found	The total number of requests with the returned HTTP status code 404.	N/A
Percentage of requests with client-side errors indicating resources not found	The percentage of requests with errors indicating resources not found out of all requests.	%
Number of requests with client-side timeout errors	The total number of requests with the returned HTTP status code 408 or OSS error code RequestTimeout.	N/A
Percentage of requests with client-side timeout errors	The percentage of requests with client-side timeout errors out of all requests.	%
Number of requests with other client-side errors	The total number of requests other than the foregoing client-side error requests with the returned HTTP status code 4xx.	N/A
Percentage of requests with other client-side errors	The percentage of requests with other client-side errors out of all requests.	%
Number of successful requests	The total number of requests with the returned HTTP status code 2xx.	N/A
Percentage of successful requests	The percentage of successful requests out of all requests.	%
Number of redirected requests	The total number of requests with the returned HTTP status code 3xx.	N/A
Percentage of redirected requests	The percentage of redirected requests out of all requests.	%

Metrics for maximum latency of OSS

Metric	Description	Unit
Maximum end-to-end latency of GetObject requests	The maximum end-to-end latency of successful GetObject requests.	ms
Maximum server latency of GetObject requests	The maximum server latency of successful GetObject requests.	ms
Maximum end-to-end latency of HeadObject requests	The maximum end-to-end latency of successful HeadObject requests.	ms

Metric	Description	Unit
Maximum server latency of HeadObject requests	The maximum server latency of successful HeadObject requests.	ms
Maximum end-to-end latency of PutObject requests	The maximum end-to-end latency of successful PutObject requests.	ms
Maximum server latency of PutObject requests	The maximum server latency of successful PutObject requests.	ms
Maximum end-to-end latency of PostObject requests	The maximum end-to-end latency of successful PostObject requests.	ms
Maximum server latency of PostObject requests	The maximum server latency of successful PostObject requests.	ms
Maximum end-to-end latency of AppendObject requests	The maximum end-to-end latency of successful AppendObject requests.	ms
Maximum server latency of AppendObject requests	The maximum server latency of successful AppendObject requests.	ms
Maximum end-to-end latency of UploadPart requests	The maximum end-to-end latency of successful UploadPart requests.	ms
Maximum server latency of UploadPart requests	The maximum server latency of successful UploadPart requests.	ms
Maximum end-to-end latency of UploadPartCopy requests	The maximum end-to-end latency of successful UploadPartCopy requests.	ms
Maximum server latency of UploadPartCopy requests	The maximum server latency of successful UploadPartCopy requests.	ms

Metrics for successful request category of OSS

Metric	Description	Unit
Number of successful GetObject requests	The number of successful GetObject requests.	N/A
Number of successful HeadObject requests	The number of successful HeadObject requests.	N/A
Number of successful PostObject requests	The number of successful PostObject requests.	N/A
Number of successful AppendObject requests	The number of successful AppendObject requests.	N/A
Number of successful UploadPart requests	The number of successful UploadPart requests.	N/A
Number of successful UploadPartCopy requests	The number of successful UploadPartCopy requests.	N/A
Number of successful DeleteObject requests	The number of successful DeleteObject requests.	N/A

Metric	Description	Unit
Number of successful DeleteObjects requests	The number of successful DeleteObjects requests.	N/A

Metrics for KVStore for Redis

Metric	Description	Apsara Stack service	Unit
CPU utilization	The CPU utilization of a KVStore for Redis instance.	KVStore for Redis	%
Memory usage	The percentage of memory that is in use.	KVStore for Redis	%
Used memory	The amount of memory that is in use.	KVStore for Redis	byte
Number of used connections	The total number of client connections that are in use.	KVStore for Redis	N/A
Percentage of used connections	The percentage of connections that are in use.	KVStore for Redis	%
Write bandwidth	The write traffic per second.	KVStore for Redis	byte/s
Read bandwidth	The read traffic per second.	KVStore for Redis	byte/s
Number of failed operations per second	The number of failed operations on a KVStore for Redis instance per second.	KVStore for Redis	count/s
Write bandwidth usage	The percentage of total bandwidth used by write operations.	KVStore for Redis	%
Read bandwidth usage	The percentage of total bandwidth used by read operations.	KVStore for Redis	%
Used QPS	The number of queries per second (QPS).	KVStore for Redis	count/s
QPS usage	The QPS usage.	KVStore for Redis	%
Average response time	The average response time.	KVStore for Redis	ms
Maximum response time	The maximum response time.	KVStore for Redis	ms
Number of failed commands	The number of failed commands.	KVStore for Redis	N/A
Hit Rate	The current hit rate.	KVStore for Redis	%
Inbound traffic	The inbound traffic to a KVStore for Redis instance.	KVStore for Redis	byte
Inbound bandwidth usage	The inbound bandwidth usage of a KVStore for Redis instance.	KVStore for Redis	%
Outbound traffic	The outbound traffic from a KVStore for Redis instance.	KVStore for Redis	byte
Outbound bandwidth usage	The outbound bandwidth usage of a KVStore for Redis instance.	KVStore for Redis	%

Metrics for VPN Gateway

Metric	Dimension	Monitoring period	Unit
Number of inbound packets in a connection per second	User and instance	1 minute	pps
Number of outbound packets in a connection per second	User and instance	1 minute	pps
Inbound bandwidth of a connection	User and instance	1 minute	bit/s
Outbound bandwidth of a connection	User and instance	1 minute	bit/s
Number of connections	User and instance	1 minute	N/A

Metrics for AnalyticDB for PostgreSQL

Metric	Description	Unit
Connection usage	The number of connections between an application and an AnalyticDB for PostgreSQL instance per second.	%
CPU utilization	The CPU utilization of an AnalyticDB for PostgreSQL instance.	%
Disk usage	The disk usage of an AnalyticDB for PostgreSQL instance.	%
IOPS usage	The number of I/O requests for an AnalyticDB for PostgreSQL instance per second.	%
Memory usage	The memory usage of an AnalyticDB for PostgreSQL instance.	%

Metrics for ApsaraDB for MongoDB

Tab	Metric	Description	Unit
	CPU utilization	The CPU utilization of an ApsaraDB for MongoDB instance.	%
	Memory usage	The memory usage of an ApsaraDB for MongoDB instance.	%
	Disk usage	The disk usage of an ApsaraDB for MongoDB instance.	%

Tab	Metric	Description	Unit
Basic metric	IOPS usage	The percentage of the IOPS used by an ApsaraDB for MongoDB instance out of the maximum available IOPS.	%
	Connection usage	The number of connections between an application and an ApsaraDB for MongoDB instance per second.	%
	QPS	The number of queries per second.	N/A
	Number of used connections	The number of current connections to an ApsaraDB for MongoDB instance.	N/A
Disk capacity	Disk space occupied by an instance	The total used space.	byte
	Disk space occupied by data	The disk space occupied by data.	byte
	Disk space occupied by logs	The disk space occupied by logs.	byte
Network request	Inbound traffic to the internal network	The inbound traffic.	byte
	Outbound traffic from the internal network	The outbound traffic.	byte
	Number of requests	The number of processed requests.	N/A
Number of operations	Number of Insert operations	None	N/A
	Number of Query operations	None	N/A
	Number of Update operations	None	N/A
	Number of Delete operations	None	N/A
	Number of Getmore operations	None	N/A
	Number of Command operations	None	N/A

Metrics for EIP

Metric	Description	Dimension	Monitoring period	Unit
Inbound bandwidth	The traffic that passes through EIP to ECS per second.	Instance	1 minute	bit/s
Outbound bandwidth	The traffic that passes through EIP from ECS per second.	Instance	1 minute	bit/s
Number of inbound packets per second	The number of packets that pass through EIP to ECS per second.	Instance	1 minute	pps
Number of outbound packets per second	The number of packets that pass through EIP from ECS per second.	Instance	1 minute	pps
Packet loss rate due to throttling	The packet loss rate when the actually used bandwidth exceeds the configured upper limit.	Instance	1 minute	pps

Metrics for API Gateway

Metric	Description	Dimension	Unit	Monitoring period
Error distribution	The number of 2xx, 4xx, and 5xx status codes returned for an API in the monitoring period.	User and API	N/A	1 minute
Inbound traffic	The total traffic of requests received by an API in the monitoring period.	User and API	byte	1 minute
Outbound traffic	The total traffic of responses sent by an API in the monitoring period.	User and API	byte	1 minute
Response time	The latency between the time when API Gateway calls the backend service of an API and the time when the result is received from the backend service in the monitoring period.	User and API	s	1 minute

Metric	Description	Dimension	Unit	Monitoring period
Number of total requests	The total number of requests received by an API in the monitoring period.	User and API	N/A	1 minute

1.1.6.2.3. View monitoring charts

You can view monitoring charts to obtain up-to-date information about each instance.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, click **Cloud Service Monitoring**.
4. Click a cloud service.
5. Click **Monitoring Charts** in the **Actions** column corresponding to an instance.

   On the Monitoring Charts page that appears, you can select a date and time to view the monitoring data of each metric.

1.1.6.3. Alerts

1.1.6.3.1. View alarm overview

On the **Overview** page in CloudMonitor, you can view the alarm status statistics and alarm logs.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, click **Overview**.
4. On the **Overview** page, view the alarm status statistics and alarm logs generated in the last 24 hours.

1.1.6.3.2. Enable or disable alert notification

You can choose whether to enable alert notification by SMS, email, or DingTalk.

Prerequisites

Valid contact information is specified when you create a user. If your contact information is changed, you must modify personal information. For more information, see Modify personal information.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **User Information**.
3. In the **Notification By** section, select **SMS**, **Email**, or **DingTalk** to enable alert notification.

   To disable alert notification, you can clear the corresponding check box.

1.1.6.3.3. View alert logs

You can view alert information to stay up to date on the running status of ECS, ApsaraDB RDS, SLB, KVStore for Redis, VPN Gateway, AnalyticDB for PostgreSQL, ApsaraDB for MongoDB, EIP, API Gateway, and OSS.

Context

Alert information contains information for all items that do not comply with your configured alert rules.

> Note
> - The system can retain up to one million alert items generated within the last three months.
> - This topic describes how to view alert information for ECS. You can view the alert information for other cloud resources in a similar manner.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, choose **Alerts > Alert History**.
4. On the **Alert Rule History List** page, filter alert information by rule ID, rule name, service, metric, and date.

   The following table describes the fields in the query result. Alert information fields

Field	Description
Product	The service for which the alert was triggered.
Fault Instance	The instance for which the alert was triggered.
Occurred At	The time when the alert was triggered.
Rule Name	The name of the alert rule.
Status	The status of the alert rule.
Notification Contact	The recipient of the alert notification.

1.1.6.3.4. Alert rules

1.1.6.3.4.1. View alert rules

After you create alert rules, you can view your alert rules on the Alert Rules page.

Context

The system provides alert rules for ECS, ApsaraDB RDS, SLB, OSS, KVStore for Redis, VPN Gateway, AnalyticDB for PostgreSQL, ApsaraDB for MongoDB, EIP, and API Gateway.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, click **Cloud Service Monitoring**.
4. Click a cloud service.
5. Click **Alert Rules** in the **Actions** column corresponding to an instance.

   On the **Alert Rules** page, view the detailed information of alert rules.

1.1.6.3.4.2. Create an alert rule

You can create an alert rule to monitor an instance.

Prerequisites

For ECS instances, you must install a monitoring plug-in to collect metric data at the operating system level.

The installation methods are as follows:

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane, choose **Cloud Service Monitoring > ECS**.
4. In the ECS instance list, select the instances that you want to monitor, and click **Batch Install**.

> **Note**
> The monitoring chart displays monitoring data 5 to 10 minutes after the monitoring plug-in is installed.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the Cloud Monitor page, click **Cloud Service Monitoring**.
4. Click a cloud service.
5. Click **Alert Rules** in the **Actions** column corresponding to an instance.

   > **Note** You can also use the search function to query specific instances for which you want to create alert rules.

6. On the **Alert Rules** page, click **Create Alert Rule**.

   Parameters for creating an alert rule

Parameter	Description
Product	The monitored cloud product.
Resource Range	The range of resources that is associated with the alert rule.
Rule Description	The description of the alert rule.
Add Rule Description	Click **Add Rule Description** to go to the rule configuration panel. For more information, see Parameters for adding rule description.
Effective Time	Only a single alert is sent during each mute duration, even if the metric value exceeds the alert rule threshold several times in a row.
Effective Period	An alert is sent only when the threshold is crossed during the effective period.
HTTP CallBack	The callback URL when the alert conditions are met.
Alert Contact Group	The group to which alerts are sent.

   Parameters for adding rule description

Parameter	Description
Rule Name	The name of the alert rule. The name must be 1 to 64 characters in length and can contain letters and digits.
Metric Name	Different products have different monitoring metrics. For more information, see Metrics.
Comparison	The comparison between thresholds and observed values. The comparison operators include >, >=, <, and <=. When the comparison rule is satisfied, an alert rule is triggered.
Threshold And Alert Level	Different metrics have different reference thresholds.

7. Click **OK**.

1.1.6.3.4.3. Disable an alarm rule

You can disable one or more alarm rules as needed.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, click **Cloud Service Monitoring**.
4. Click a cloud service.
5. Click **Alarm Rules** in the **Actions** column corresponding to an instance to go to its **Alarm Rules** page.
6. Select the alarm rule that you want to disable, and click **Disable** below the alarm rule list.
7. In the message that appears, click **OK**.

1.1.6.3.4.4. Enable an alarm rule

After an alarm rule is disabled, it can be re-enabled as needed.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, click **Cloud Service Monitoring**.
4. Click a cloud service.
5. Click **Alarm Rules** in the **Actions** column corresponding to an instance to go to its **Alarm Rules** page.
6. Select the alarm rule that you want to enable, and click **Enable** below the alarm rule list.
7. In the message that appears, click **OK**.

1.1.6.3.4.5. Delete an alarm rule

You can delete alarm rules that are no longer needed.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, click **Cloud Service Monitoring**.

4. Click a cloud service.
5. Click **Alarm Rules** in the **Actions** column corresponding to an instance to go to its **Alarm Rules** page.
6. Select the alarm rule that you want to delete and click **Delete** in the **Actions** column.
7. In the message that appears, click **OK**.

1.1.7. VMware Cloud on Alibaba Cloud

1.1.7.1. VMware Cloud on Alibaba Cloud

1.1.7.1.1. Log on to the VMware Cloud on Alibaba Cloud console

This topic describes how to log on to the VMware Cloud on Alibaba Cloud console.

Prerequisites

- Before you log on to the Apsara Uni-manager Management Console, you must obtain the endpoint of the console from the deployment personnel.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products > Elastic Computing > VMware Cloud on Alibaba Cloud**.

1.1.7.1.2. Bind a VMware Cloud on Alibaba Cloud region

Before you use VMware Cloud on Alibaba Cloud, you must bind a VMware Cloud on Alibaba Cloud region to an organization.

Prerequisites

A VMware Cloud on Alibaba Cloud region is managed. For more information, see Add a VMware node.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**.

3. In the left-side navigation pane, click **Resource Pools**.
4. In the organization navigation tree, click an organization. In the **Regions** section, select the region that you want to bind.
5. Click **Update Association**.

1.1.7.1.3. Instructions

1.1.7.1.3.1. Limits

Before you use VMware Cloud on Alibaba Cloud virtual machine (VM) templates, you must familiarize yourself with the limits of instances.

General limits

- You must select appropriate operating systems for VMware Cloud on Alibaba Cloud VM templates.

  The following operating systems are verified to be available in the Apsara Uni-manager Management Console:

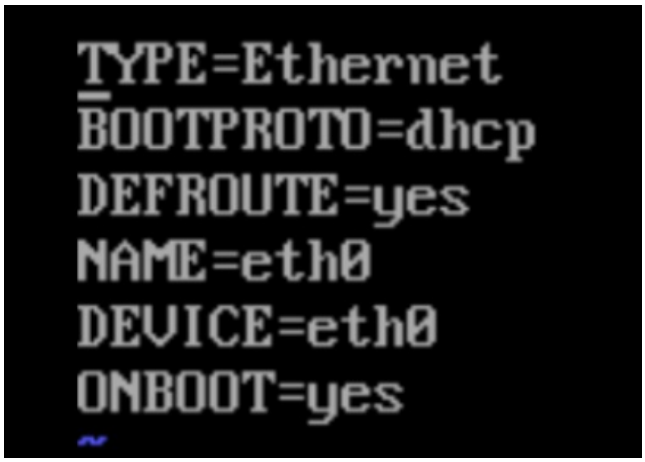
  - CentOS8.2.2004
  - CentOS7.2003
  - CentOS6.10
  - Ubuntu-20.04.1
  - Ubuntu-18.04.5
  - Ubuntu-16.04.7
  - Windows Server 2016
  - Windows Server 2019

- The Apsara Uni-manager Management Console supports VMware vSphere 6.x. Other versions of VMware vSphere, such as 5.x or 7.x, can in theory be supported. However, the specific support depends on the compatibility of the VMware Cloud on Alibaba Cloud API and must be evaluated by the R&D team of the Apsara Uni-manager Management Console.
- You must install VMware Tools.

  For more information, see the VMware documentation. Select Full Installation in the installation process.
- You must modify network interface controller configurations in the operating system of the VM.

  When you create a VM in the Apsara Uni-manager Management Console, you can specify the IP address of the operating system. This feature is supported by valid network interface controller configurations.

Operating systems of VM templates must be in DHCP mode. Information such as the MAC address and universally unique identifier (UUID) in the network interface controller configurations must be removed. The following information can be retained.



> **Note**
>
> Some configurations are required for the following operating systems:
>
> - CentOS 6: You must clear the content in the network interface controller configuration file named 70-persistent-net.rules. The file is stored in the /etc/udev/rules.d/ directory.
> - CentOS 7: The system generates the name for a network interface controller, such as ifcfg-ens160. You must modify the name to ifcfg-eth0 to make the name take effect.
> - Ubuntu18.04, 20.04, and later: You must run the `sudo rm /etc/netplan/*.yaml` command to remove the network interface controller configurations.

1.1.7.1.3.2. Suggestions

Consider the following operation suggestions to make more efficient use of VMware Cloud on Alibaba Cloud virtual machine (VM) templates.

- Select the latest version of VM hardware.
- Select thin provision for VM disks.

  Disk replication is required when you create VMs based on templates. Files of disks of the thin provision type are small in size. This can help accelerate the creation of VMs.

> **Note**
>
> Large sizes of disk files in VM templates or slow storage write speeds may cause VM creation to time out and fail. The maximum timeout period supported by the Apsara Uni-manager Management Console is 10 minutes.

1.1.7.1.4. Instances

1.1.7.1.4.1. Create a VMware Cloud on Alibaba Cloud instance

A VMware Cloud on Alibaba Cloud instance is a virtual machine (VM) that contains the basic computing components of a server, such as CPU, memory, operating system, network, and disks.

Prerequisites

- The region where VMware Cloud on Alibaba Cloud is deployed is managed. For more information, see Add a VMware node.
- The region where VMware Cloud on Alibaba Cloud is deployed is bound to an organization. For more information, see Bind a VMware Cloud on Alibaba Cloud region.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Click **Create Instance** in the upper-right corner.
5. Configure parameters listed in the following table to create an instance.

Section	Parameter	Required	Description
Basic Settings	**Organization**	Yes	The organization in which to create the instance.
	Resource Set	Yes	The resource set in which to create the instance.
Region	**Region**	Yes	The region in which to create the instance.
	Zone	Yes	The zone in which to create the instance.
	VPC	Yes	The VPC in which to create the instance.
	vSwitch	Yes	Select the vSwitch to which the instance belongs. The vSwitch corresponds to the port group of a VMware ESXi host or a distributed switch, and maps to the VLAN of a physical switch.
	Private IP Address	Yes	The private IPv4 address of the instance. The private IPv4 address must be within the CIDR block of the vSwitch.
	Private Subnet Mask	Yes	The private subnet mask. Example: 255.255.255.0. The specified subnet mask must be within the CIDR block of the selected vSwitch.

Section	Parameter	Required	Description
	Private IP Address of Gateway	Yes	The private IP address of the gateway. Example: 192.168.100.1. The IP address must be within the CIDR block of the selected vSwitch.
	Private IP Address of DNS Server	No	The private IP address of the DNS server. Example: 114.114.114.114. The IP address must be within the CIDR block of the selected vSwitch.
Instance	**Instance Family**	No	The instance family of the instance. Valid values: ◦ Memory Optimized ◦ Compute Optimized ◦ General Purpose
	Instance Type	Yes	The instance type of the instance. You can specify the vCPUs and memory.
Image	**Image Type**	No	The type of the image. Default value: **Public Image**.
	Public Image	Yes	The public image of the instance.
	System Disk (GB)	No	The system disk to which the operating system is installed. You can configure different storage types for the disk. Valid values: ◦ **Shared Storage: All**: The system selects an available shared storage. We recommend that you select this type. ◦ **Shared Storage: storageA**: The storage named storageA of the VMware Cloud on Alibaba Cloud instance is used. Administrators must make sure the storage is appropriate. If the storage capacity is insufficient, the instance fails to be created.

Section	Parameter	Required	Description
Storage	**Data Disk (GB)**	No	You can also add data disks after the instance is created. You can configure different storage types for the disk. Valid values: ◦ **Shared Storage: All**: The system selects an available shared storage. We recommend that you select this type. ◦ **Shared Storage: storageA**: The storage named storageA of the VMware Cloud on Alibaba Cloud instance is used. Administrators must make sure the storage is appropriate. If the storage capacity is insufficient, the instance fails to be created. You must also specify the provision type when you create the instance. Valid values: ◦ Thin Provision: Storage space increases with the use of the disk. ◦ Thick Provision Lazy Zeroed: Storage space is equal to the size of the disk and does not increase. The disk is formatted when data is written. ◦ Thick Provision Eager Zeroed: Storage space is equal to the size of the disk and does not increase. The storage of the disk is immediately formatted when the disk is created.
Password	**Password Setting**	No	Select **Set after Purchase**.
Instance Name	**Instance Name**	Yes	The name of the instance. The name must be 2 to 128 characters in length and can contain letters, periods (.), underscores (_), hyphens (-), and colons (:). It must start with a letter and cannot start with http:// or https://.

6. Click **Submit**.

1.1.7.1.4.2. View instance information

You can view the list of created instances as well as details of individual instances, such as their basic configurations, disks, and elastic network interfaces (ENIs).

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.

   You can view the list of VMware Cloud on Alibaba Cloud instances that are deployed in the current region.
4. Use one of the following methods to go to the details page of an instance:
   - In the **Instance ID/Name** column, click the instance ID.
   - Click **Manage** in the **Actions** column corresponding to the instance.
   - Choose **More** > **Show Details** in the **Actions** column corresponding to the instance.

1.1.7.1.4.3. Modify an instance

You can modify the name and description of a created VMware Cloud on Alibaba Cloud instance.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Find the instance that you want to modify and choose **More** > **Modify** in the **Actions** column.
5. Modify the name and description of the instance.
6. Click **OK**.

1.1.7.1.4.4. Remotely connect to an instance

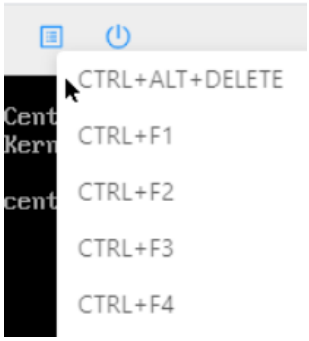
You can remotely connect to and manage added VMware Cloud on Alibaba Cloud instances.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Find the instance that you want to manage and click **Remote Connection** in the **Actions** column.
5. Enter the username and password.
   - For a Linux instance, enter the username *root* and the logon password.

> **Note**
>
> When you log on to the Linux instance, the password is not displayed as you enter it. Press the Enter key after you enter the password.

- For a Windows instance, to use a key combination such as Ctrl+Alt+Delete, click the List icon in the upper-right corner of the page and select the corresponding composite key from the drop-down list.



Enter the username and password, and click the Log On icon.

1.1.7.1.4.5. Stop an instance

You can stop VMware Cloud on Alibaba Cloud instances that are not in use. The stop operation interrupts services that are running on the instances. Exercise caution when you perform this operation.

Prerequisites

The instance is in the **Running** state.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Use one of the following methods to stop the instance:
   - To stop a single instance, find the instance and choose **More** > **Instance Status** > **Stop** in the **Actions** column.
   - To stop one or more instances at a time, select the instances and click **Stop** in the lower part of the Instances page.
5. Click **OK**.

Execution results

When the instance is being stopped, its status in the **Status** column changes from **Running** to **Stopping**. After the instance is stopped, its status changes to **Stopped**.

1.1.7.1.4.6. Start an instance

You can start a stopped instance.

Prerequisites

The instance is in the **Stopped** state.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Use one of the following methods to start the instance:
   - To start a single instance, find the instance and choose **More** > **Instance** > **Status** > **Start** in the **Actions** column.
   - To start one or more instances at a time, select the instances and click **Start** in the lower part of the Instances page.
5. Click **OK**.

Execution results

When the instance is being started, its status in the **Status** column changes from **Stopped** to **Starting**. After the instance is started, its status changes to **Running**.

1.1.7.1.4.7. Restart an instance

After you change the logon password of an instance or install system updates, you must restart the instance. The restart operation stops the instances for a period of time. This causes the services that are running on the instances to be interrupted. Exercise caution when you perform this operation.

Prerequisites

The instance is in the **Running** state.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Use one of the following methods to restart the instance:
   - To restart a single instance, find the instance and choose **More** > **Instance Status** > **Restart** in the **Actions** column.
   - To restart one or more instances at a time, select the instances and click **Restart** in the lower part of the Instances page.
5. In the Restart Instance dialog box, select a restart mode.
   - **Restart**: restarts the instance normally.
   - **Force Restart**: forces the instance to restart. This may result in loss of unsaved data.
6. Click **OK**.

1.1.7.1.4.8. Delete an instance

You can delete instances that are no longer needed to release their resources. Deleted instances cannot be recovered. We recommend that you back up data before you delete an instance. If data disks are released with the instances, the disk data cannot be recovered.

Prerequisites

The instance is in the **Stopped** state.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Select the instance and click **Delete** in the lower part of the Instances page.
5. Click **OK**.

1.1.7.1.5. Images

1.1.7.1.5.1. Create a custom image

You can create a custom image and use it to create identical instances or replace the system disks of existing instances. This way, you can configure many instances that have identical operating systems and data environments.

Create a custom image from an instance

You can create a custom image from an instance to replicate the data of all system and data disks on the instance.

> **Note**
>
> To avoid data security risks, we recommend that you delete sensitive data from an instance before you use the instance to create a custom image.

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Find the instance from which you want to create a custom image and choose **More** > **Create Custom Image** in the **Actions** column.
5. Set the name, sharing scope, and description for the custom image, and click **OK**.

   The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), hyphens (-), and colons (:). It cannot start with a special character or digit.

   You can set the sharing scope to the permission scope of the image.

   The description must be 2 to 256 characters in length and cannot start with http:// or https://.

1.1.7.1.5.2. View images

You can view the list of created images.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, choose **Images** > **Images**.
3. In the top navigation bar, move the pointer over **Region** and select the region where the image is created.

4. Select a filter option, enter the corresponding information in the search box, and then click **Search**.

   You can select multiple filter options to narrow down search results.

Parameter	Description
Image Name	The image name used to search for the image.
Image ID	The image ID used to search for the image.

1.1.7.1.6. Snapshots

1.1.7.1.6.1. Create a snapshot

You can manually create a snapshot for a disk to back up disk data.

Prerequisites

- The instance to which the disk is attached is in the **Running** or **Stopped** state.
- The disk is in the **Running** state.

Background information

A snapshot of a disk can be used to roll back data of the disk.

When you create a snapshot, take note of the following items:

- For each disk, the first snapshot is a full snapshot and subsequent snapshots are incremental snapshots. It takes an extended period of time to create the first snapshot. It takes a short period of time to create an incremental snapshot. The amount of taken time depends on the volume of data that has been changed since the latest snapshot. The more data that has been changed, the more time it takes.
- Avoid creating snapshots during peak hours.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Find the instance from which you want to create a snapshot and click the **Snapshots** tab.
5. Click **Create and Bind Snapshot**.
6. Set the name, type, and description for the snapshot, and click **Submit**.

Parameter	Description
Snapshot Name	The name of the snapshot.

Parameter	Description
Snapshot Type	The type of the snapshot. Valid values: ○ Disk Snapshot ○ Memory Snapshot
Snapshot Description	The description of the snapshot.

1.1.7.1.6.2. Delete a snapshot

You can delete a snapshot that is no longer needed. After the snapshot is deleted, it cannot be recovered.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Find the instance whose snapshot is to be deleted and click the **Snapshots** tab.
5. Use one of the following methods to delete the snapshot:
   - To delete a single snapshot, find the snapshot and click **Delete** in the **Actions** column.
   - To delete one or more snapshots at a time, select the snapshots and click **Delete** in the lower part of the Snapshots tab.
6. Click **OK**.

1.1.7.1.6.3. View snapshots

You can view the list of created snapshots.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Find the instance in which you want to view snapshots and click the **Snapshots** tab.
5. Select a filter option, enter the corresponding information in the search box, and then click **Search**.

   You can select multiple filter options to narrow down search results.

Parameter	Description
Snapshot Name	The snapshot name used to search for the snapshot.
Snapshot ID	The snapshot ID used to search for the snapshot.

1.1.7.1.7. Disks

1.1.7.1.7.1. Create a disk

To increase the storage space of VMware Cloud on Alibaba Cloud instances, you can create standalone data disks and then attach them to the instances. This topic describes how to create an empty data disk. You cannot create standalone system disks.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Find the instance for which you want to create a disk and click the **Disks** tab.
5. Click **Create and Attach Disk**.
6. Configure parameters listed in the following table to create a disk.

Section	Parameter	Required	Description
Region	**Zone**	Yes	The zone in which to create the disk.
Basic Settings	**Specifications**	Yes	The disk category and the disk size.
	Provision Type	Yes	The provision type. Valid values: ◦ Thin Provision: Storage space increases with the use of the disk. ◦ Thick Provision Lazy Zeroed: Storage space is equal to the size of the disk and does not increase. The disk is formatted when data is written. ◦ Thick Provision Eager Zeroed: Storage space is equal to the size of the disk and does not increase. The storage of the disk is immediately formatted when the disk is created.

7. Click **Submit**.

Execution results

The created disk is displayed in the disk list and in the **Running** state.

1.1.7.1.7.2. View disks

You can view the list of created disks and the details of individual disks.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Find the instance for which you want to view disks and click the **Disks** tab.
5. Select a filter option from the drop-down list, enter the relevant information in the search box, and then click **Search**.

   You can select multiple filter options to narrow down search results.

Parameter	Description
Disk Name	The disk name used to search for the disk.
Disk ID	The disk ID used to search for the disk.
Disk Properties	The disk type used to search for disks of that type. Valid values: ○ All ○ System Disk ○ Data Disk

1.1.7.1.7.3. Detach a data disk

You can detach data disks. System disks cannot be detached.

Procedure

> **Warning**
>
> Resources are released after disks are detached. Make sure that the data of a disk is backed up before you detach it.

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Find the instance from which you want to detach a data disk and click the **Disks** tab.

5. Find the data disk that you want to detach and choose **More** > **Detach** in the **Actions** column.
6. Click **OK**.

1.1.7.1.8. ENIs

1.1.7.1.8.1. Create an ENI

You can create and bind elastic network interfaces (ENIs) to VMware Cloud on Alibaba Cloud instances.

Prerequisites

A virtual private cloud (VPC) and a vSwitch are created. For more information, see Create a VPC and Create a vSwitch in *Apsara Stack VPC User Guide*.

Background information

ENIs are classified into primary and secondary ENIs.

A primary ENI is created by default when an instance is created in a VPC. This primary ENI has the same lifecycle as the instance and cannot be unbound from the instance.

ENIs that are separately created are secondary ENIs. This topic describes how to create a secondary ENI.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Find the instance for which you want to create an ENI and click the **ENIs** tab.
5. Click **Create and Bind ENI**.
6. Configure parameters listed in the following table to create an ENI.

Section	Parameter	Required	Description
Region	**Organization**	Yes	The organization in which to create the ENI.
	Resource Set	Yes	The resource set in which to create the ENI.
	Region	Yes	The region in which to create the ENI.
	Zone	Yes	The zone in which to create the ENI.

Section	Parameter	Required	Description
Basic Settings	**VPC**	Yes	The VPC in which to create the ENI. The secondary ENI can be bound only to an instance in the same VPC. **Note** After the ENI is created, you cannot change its VPC.
	vSwitch	Yes	The vSwitch to be associated with the ENI. The secondary ENI can be bound only to an instance in the same VPC. Select a vSwitch that is deployed within the same zone as the instance to which the ENI is bound. The vSwitch of the ENI can be different from that of the instance. **Note** After the ENI is created, you cannot change its vSwitch.

7. Click **Submit**.

Execution results

The created ENI is displayed on the ENIs page and is in the **Bound** state.

1.1.7.1.8.2. View ENIs

You can view the list of created elastic network interfaces (ENIs).

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Find the instance for which you want to view ENIs and click the **ENIs** tab.
5. Select a filter option from the drop-down list, enter the relevant information in the search box, and then click **Search**.

   You can select multiple filter options to narrow down search results.

Parameter	Description
ENI Name	The ENI name used to search for the ENI.
ENI ID	The ENI ID used to search for the ENI.
VSwitch ID	The vSwitch ID used to search for the ENIs that are associated with the vSwitch.

1.1.7.1.8.3. Delete an ENI

You can delete secondary elastic network interfaces (ENIs) that are no longer needed.

Background information

Only secondary ENIs can be deleted. Primary ENIs share the same lifecycle as instances and cannot be deleted.

Procedure

1. Log on to the VMware Cloud on Alibaba Cloud console.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, move the pointer over **Region** and select the region where VMware Cloud on Alibaba Cloud is deployed.
4. Find the instance whose secondary ENI is to be deleted and click the **ENIs** tab.
5. Find the secondary ENI and click **Delete** in the **Actions** column.
6. Click **OK**.

1.1.8. Enterprise

1.1.8.1. Organizations

1.1.8.1.1. Create an organization

You can create organizations to store resource sets and their resources.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the Enterprise page, click **Organizations**.
4. In the organization navigation tree, click a parent organization. In the Current Organization section, click **Add Organization**.
5. In the Create Organization dialog box, enter an organization name and click **OK**.

1.1.8.1.2. Query an organization

You can query an organization by name to view its resource sets, users, and user groups.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the search box below **Organizations**, enter an organization name to query information about the corresponding organization.

1.1.8.1.3. View organization information

You can view information about an organization on the Organizations page.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. On the **Organizations** page, click an organization in the organization list.
5. On the right side of the page, view the organization information.
   - In the **Resource Sets** section, you can view information such as the name, creation time, and creator of each resource set in the organization. Click the name of a resource set to view its details.
   - In the **Users** section, you can view information such as the name, status, and role of each user in the organization. Click a username to view the user details.
   - In the **User Groups** section, you can view the name, organization, role, users, and creation time of each user group in the organization.

1.1.8.1.4. Modify the name of an organization

Users that have operation permissions on an organization can modify the name of the organization.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the organization navigation tree, click an organization name.
5. In the Current Organization section, click **Edit Organization**.
6. In the Edit Organization dialog box, modify the organization name.
7. Click **OK**.

1.1.8.1.5. Change organization ownership

Users that have operation permissions on organizations can change the ownership of organizations.

Prerequisites

- Make sure that each organization under the target organization has a unique name.
- The ownership of an organization cannot be changed cross level-1 organizations.

Context

Users can change the ownership of an organization cross parent organizations. This way, the ownership of subordinate organizations, users, and resources are also changed in a cascading manner.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Change Ownership**.
4. On the **Change Ownership** page, select the target organization and click **Change Ownership** on the right.
5. In the **Change Organization** dialog box, select the destination organization and click **OK** to change the ownership of the target organization and resources sets and users under this organization.

1.1.8.1.6. Obtain the AccessKey pair of an organization

An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey pair is used to implement symmetric encryption to verify the identity of the requester. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt the signature string. This topic describes how to obtain the AccessKey pair of an organization.

Prerequisites

Only operations administrators and level-1 organization administrators can obtain the AccessKey pair of an organization.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, click **Organizations**.
4. In the organization navigation tree, click an organization name.
5. In the Current Organization section, click **Obtain AccessKey Pair**.
6. In the AccessKey message, view the AccessKey pair of the organization.

1.1.8.1.7. Delete an organization

Administrators can delete organizations that are no longer needed.

Prerequisites

Before you delete an organization, make sure that the organization does not contain users, resource sets, or subordinate organizations. Otherwise, the organization cannot be deleted.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the Enterprise page, click **Organizations**.
4. In the organization navigation tree, click an organization name. In the **Current Organization** section, click **Delete Organization**.
5. In the Confirm message, click **OK**.

1.1.8.2. Resource sets

1.1.8.2.1. Create a resource set

You must create a resource set before you apply for resources.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Resource Sets**.
4. In the upper-left corner of the **Resource Sets** page, click **Create Resource Set**.
5. In the **Create Resource Set** dialog box, set **Name** and **Organization**.
6. Click **OK**.

1.1.8.2.2. View the details of a resource set

When you want to use a cloud resource in your organization, you can view the details of the resource set that contains the resource, including all resource instances and users of the resource set.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Resource Sets**.
4. Select an **organization** from the drop-down list, or enter a **resource set** name in the search bar, and then click **Search**.
5. Click the name of the target **resource set**.
6. On the **Resource Set Details** page, click the **Resources** and **Members** tabs to view information about all resource instances and users of the resource set.
7. On the **Resources** tab, click the number of a service to go to the instance list page of the service. The list is automatically filtered and displayed based on the organization and resource set.

1.1.8.2.3. Modify the name of a resource set

An administrator can modify the name of a resource set to keep it up-to-date.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Resource Sets**.
4. Click **More** in the **Actions** column corresponding to a resource set, and choose **Edit Name** from the shortcut menu.
5. In the dialog box that appears, enter the new name.
6. Click **OK**.

1.1.8.2.4. Add a member to a resource set

You can add a member to a resource set so that the member can use the resources in the resource set.

Prerequisites

Before adding a member, make sure that the following prerequisites are met:

- A resource set is created. For more information, see Create a resource set.
- A user is created. For more information, see Create a user.

Context

Members of a resource set have the permissions to use resources in the resource set.

Deleting resources from a resource set does not affect the members of the resource set. Similarly, deleting members from a resource set does not affect the resources in the resource set.

You can delete a member that is no longer in use in a resource set. After the member is deleted, it will no longer be able to access the resource set.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Resource Sets**.
4. Click **More** in the **Actions** column corresponding to a resource set, and choose **Add Member** from the shortcut menu.
5. In the dialog box that appears, select a username.
6. Click **OK**.

1.1.8.2.5. Add or remove a user group of a resource set

You can add or remove a user group of a resource set to manage user group access to resources in the resource set.

Prerequisites

- A resource set is created. For more information, see Create a resource set.
- A user group is created. For more information, see Create a user group.

Context

User groups in a resource set have the permissions to use resources in the resource set.

Deleting resources from a resource set does not affect user groups of the resource set. Similarly, deleting user groups from a resource set does not affect the resources in the resource set.

You can delete a user group that is no longer in use in a resource set. After the user group is deleted, it will no longer be able to access the resource set.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Resource Sets**.
4. Click **More** in the **Actions** column corresponding to the target resource set.
5. Add or remove a user group.
   - Select **Add User Group**. In the dialog box that appears, select a user group. Click **OK** to add the user group.
   - Select **Delete User Group**. In the dialog box that appears, select a user group. Click **OK** to remove the user group.

1.1.8.2.6. Delete a resource set

You can delete resource sets that are not needed as an administrator.

Prerequisites

Ensure that the resource set to be deleted does not contain resources, users, or user groups.

> **Notice** A resource set cannot be deleted if it contains resources, users, or user groups.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Resource Sets**.
4. Click **More** in the **Actions** column corresponding to the target resource set, and select **Delete**.
5. In the message that appears, click **OK**.

1.1.8.3. Roles

1.1.8.3.1. Create a custom role

You can create custom roles in the Apsara Uni-manager Management Console to more efficiently grant permissions to users so that different personnel can work with different features.

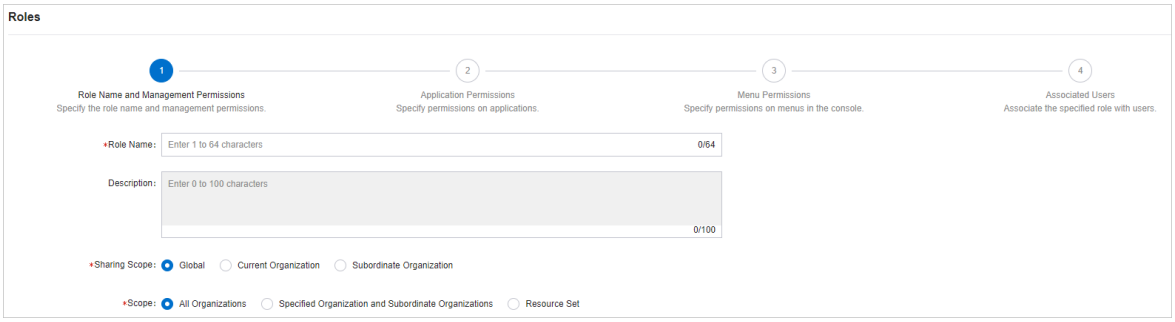
Context

A role is a set of access permissions. Each role has a range of permissions. A user can have multiple roles, which means that the user is granted all of the permissions defined for each role. A role can be used to grant the same set of permissions to a group of users.

The total number of custom and default roles cannot exceed 20.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the upper-right corner of the page, click **Create Custom Role**.
5. On the **Roles** page, set the role name and management permissions.



The following table describes the role parameters.

Role parameters

Parameter	Description
Role Name	The name of the RAM role. The name can be up to 15 characters in length and can contain only letters and digits.
Description	Optional. The description of the role. The description can be up to 100 characters in length and can contain letters, digits, commas (,), semicolons (;), and underscores (_).
Sharing Scope	○ **Global** The role is visible and valid to all organizations involved. The default value is Global. ○ **Current Organization** The role is visible and valid to the organization to which the user belongs. ○ **Subordinate Organization** The role is visible and valid to the organization to which the user belongs and its subordinate organizations.
Scope	○ **All Organizations** The permissions apply to all organizations involved. ○ **Specified Organization and Subordinate Organizations** The permissions apply to the organization to which the user belongs and its subordinate organizations. ○ **Resource Sets** The permissions apply to the resource sets that are assigned to the user.

6. Select the operation permissions that this role has and click **Next**.
7. In the **Application Permissions** step, select the operation permissions that this role has on the cloud services, and click **Next**.
8. In the **Menu Permissions** step, select the operation permissions that this role has on the menus and the homepage template corresponding to the role, and click **Create Role**.
9. In the **Associated Users** step, select the users associated with the role from the drop-down list. The associated users are granted the permissions of the role.

1.1.8.3.2. View the details of a role

If you are uncertain about the specific permissions of a role, you can go to the **Roles** page to view the role permissions.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. Click the name of the role that you want to view. On the **Roles** page, view the information of the role.

1.1.8.3.3. Modify custom role information

You can modify the name and permissions of a custom role as an administrator.

Context

Information about preset roles cannot be modified.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, click **More** in the **Actions** column corresponding to the target custom role, and select **Modify**.
5. On the **Roles** page, modify the custom role name, permissions, and associated users or user groups.
   - Modify role name: Enter a new role name in the **Role Name** field.
   - Modify permissions: Click the **Management Permissions**, **Application Permissions**, or **Menu Permissions** tab, select or clear related permissions from the corresponding tab, and then click **Update**.
   - Bind a user to a role: Click the **Associated Users** tab and select a user from the **Select one or more users** drop-down list to add the user. To unbind the user from the role, click **Remove** in the **Actions** column.
   - Manage user groups: Click the **User Groups** tab, click **Add User Group**, select a user group from the drop-down list, and then click **OK** to bind the user group. To unbind the user group from the role, click **Remove** in the **Actions** column.

1.1.8.3.4. Copy a role

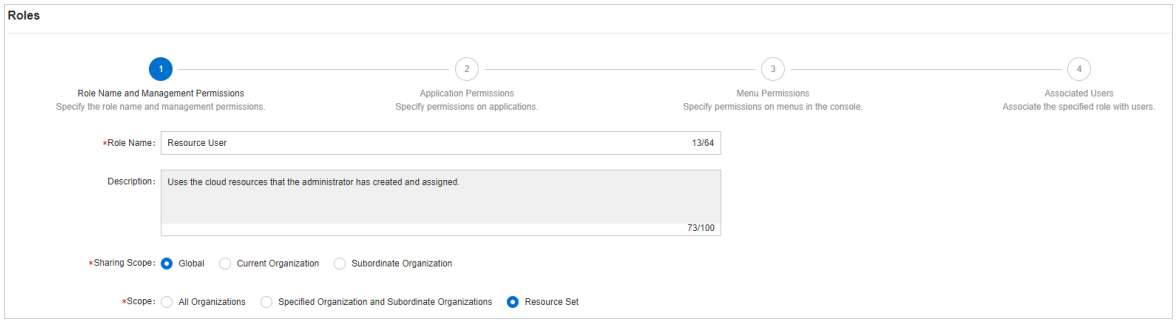
You can copy a preset role or a custom role to create a role that has the same permissions.

Context

Operations on the **Roles** page are the same as those for creating a custom role. You can add, modify, and remove the role permissions in the copied role. By default, if you do not modify the role permissions, the sharing scope, management permissions, application permissions, menu permissions, and associated users of the copied role are all the same as those of the source role.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role list, choose **More** > **Copy** in the **Actions** column corresponding to a role.
5. On the **Roles** page, set the new role name, sharing scope, and management permissions.



> **Note** The role name must be unique.

6. Select the operation permissions that this role has and click **Next**.

7. In the **Application Permissions** step, select the operation permissions that this role has on the cloud services and click **Next**.
8. In the **Menu Permissions** step, select the operation permissions that this role has on the menus and click **Create Role**.
9. In the **Associated Users** step, select the users that are associated with the role from the drop-down list. The associated users are granted the permissions of the role.

1.1.8.3.5. Disable a role

When you disable a role, the permissions of the role are disabled.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role list, click **More** in the **Actions** column corresponding to a role and choose **Disable** from the shortcut menu.

1.1.8.3.6. Enable a role

When you enable a disabled role, the permissions of the role are restored.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role list, click **More** in the **Actions** column corresponding to a disabled role and choose **Enable** from the shortcut menu.

1.1.8.3.7. Delete a custom role

You can delete a custom role that is no longer needed.

Prerequisites

- Default or preset roles cannot be deleted.
- To delete a role, you must unbind all user groups from the role.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. Choose **More** > **Delete** in the **Actions** column corresponding to a role.
5. In the Confirm message, click **OK**.

1.1.8.4. Users

1.1.8.4.1. System users

1.1.8.4.1.1. Create a user

You can create a user and assign the user different roles as an administrator to meet different requirements for system access control.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. Use one of the following methods to open the Create User window:
   - In the left-side navigation pane of the **Enterprise** page, click **Organizations**. In the **Users** section of the **Organizations** page, click **Create User**.
   - In the left-side navigation pane of the **Enterprise** page, click **Users**. On the **System Users** tab of the **Users** page, click **Create**.
4. In the Create User dialog box, configure the parameters.

Parameter	Description
Username	The Apsara Stack account name of the user. The name must be 3 to 30 characters in length, and can contain letters, digits, hyphens (-), underscores (_), periods (.), and at signs (@). It must start with a letter or digit.
Display Name	The display name of the user. The name must be 2 to 30 characters in length, and can contain letters, digits, hyphens (-), underscores (_), periods (.), and at signs (@).
Roles	The role to be assigned to the user.
Organization	The organization to which the user belongs.
Logon Policy	The logon policy that restricts the logon time and IP addresses of the user. The default policy is automatically bound to new users. **Note** The default policy does not restrict the time period and IP addresses for users to log on. To restrict the logon time and IP addresses of a user, you can modify the logon policy of the user or create a logon policy for the user. For more information, see Create a logon policy.
Mobile Number	The mobile number of the user. The mobile number is used by the system to notify users of resource application and usage. Make sure that the entered mobile number is correct. **Note** If the mobile number is changed, update it on the system in a timely manner.
Landline Number	Optional. The landline number of the user. The landline number must be 4 to 20 characters in length, and can contain only digits and hyphens (-).

Parameter	Description
Email	The email address of the user. Emails about the usage and requests for resources will be sent to the email address. Make sure that the specified email address is correct. **Note** If the email address is changed, update it on the system in a timely manner.
DingTalk Key	The key of the chatbot for the DingTalk group where the user is a member.
Notify User by SMS	After this option is selected, the Apsara Uni-manager Management Console informs the user configured as the alert contact by SMS whenever an alert is generated. **Note** You must configure an SMS server to receive an SMS message each time an alert is triggered. For more information, contact on-site O&M engineers.
Notify User by Email	After this option is selected, the Apsara Uni-manager Management Console informs the user configured as the alert contact by email whenever an alert is generated. **Note** You must configure an email server to receive an email each time an alert is triggered. For more information, contact on-site O&M engineers.
Notify User by DingTalk	After this option is selected, the Apsara Uni-manager Management Console informs the user configured as the alert contact by DingTalk whenever an alert is generated.

5. Click **OK**.

1.1.8.4.1.2. Query a user

You can view user information such as name, organization, mobile number, email address, role, logon time, and initial password.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **System Users** tab.
5. Set **Username**, **Organization**, or **Role**, and then click **Search**.
6. Click **More** in the **Actions** column corresponding to a user, and choose **User Information** from the shortcut menu to view basic information about the user.

1.1.8.4.1.3. Modify user information

You can modify user information such as display name, mobile number, and email address to keep it up to date.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.

3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **System Users** tab.
5. Find the user that you want to modify and choose **More** > **Edit** in the **Actions** column.
6. In the **Modify User Information** dialog box, enter the relevant information and click **OK**.

1.1.8.4.1.4. Change user roles

You can add, change, and delete roles for a user.

Change user roles by using user management

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **System Users** tab.
5. Choose **More** > **Authorize** in the **Actions** column corresponding to a user.
6. In the **Role** field, add, delete, or change user roles.
7. Click **OK**.

Change user roles by changing ownership

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Change Ownership**.
4. Click the ▸ icon to the left of an organization and click **Users**.
5. In the **Users** section on the right, set **Logon Policy** and **Role** or **Username**, and click **Search** to query a user.
6. Find the user and click **Change** in the **Actions** column.
7. In the **Organization to Change** dialog box, select the destination or original organization and select the role to be added or removed from the **Assigned Roles** drop-down list.

   > **Note**
   > - If you change only roles without changing the organization, select the original organization.
   > - Blue role names are the roles that are selected, and black role names are the roles that are not selected.

8. Click **OK**.

1.1.8.4.1.5. Modify the information of a user group

On the Users page, you can view the user group information and modify the ownership of users in user groups.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, click **Users**.
4. Click the **System Users** tab, select a target user, and then click **More** in the **Actions** column.
   - Select **Add to User Group**. In the dialog box that appears, select the target user group and click **OK** to add the user to the user group.

- Select **Remove from User Group**. In the dialog box that appears, select the target user group and click **OK** to remove the user from the user group.

1.1.8.4.1.6. Modify a user logon policy

An administrator can modify a user's logon policy to restrict the permitted logon time and IP addresses of the user.

Prerequisites

A new logon policy is created. For more information about how to create a logon policy, see Create a logon policy.

Modify a user logon policy

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **System Users** tab.
5. Click **More** in the **Actions** column corresponding to a user, and choose **Logon Policy** from the shortcut menu.
6. In the **Assign Logon Policy** dialog box, select a logon policy and click **OK**.

Modify multiple user logon policies at a time

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **System Users** tab.
5. Select multiple users.
6. In the upper-right corner of the page, click **Logon Policy**.
7. In the **Assign Logon Policies** dialog box, select a logon policy and click **OK**.

1.1.8.4.1.7. View the initial password of a user

After a user is created, the system generates an initial password for the user.

Context

Organization administrators can view the initial passwords of all users in the organizations they manage.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. Use one of the following methods to view the initial password of a user on the **Enterprise** page:
   - In the left-side navigation pane, click **Users**. On the **System Users** tab of the **Users** page, select a username.
     - Click **View Initial Password** in the upper-right corner of the **Users** page to view the initial password.
     - Choose **More** > **User Information** in the **Actions** column corresponding to the user. On the user information page, click **View Password** to view the initial password.
   - In the left-side navigation pane, click **Organizations**. In the organization navigation tree on the

Organizations page, click an organization name. In the **Users** section, click a username. On the user information page, click **View Password** to view the initial password.

1.1.8.4.1.8. Reset the password of a user

If users forget their logon passwords, the system administrator can reset the logon passwords for them.

Prerequisites

Only organization administrators can reset the password of a user.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. Use one of the following methods to go to the **User Information** page:
   - In the left-side navigation pane of the **Enterprise** page, click **Users**. On the **System Users** tab of the **Users** page, click a username.
   - In the left-side navigation pane of the **Enterprise** page, click **Organizations**. On the **Organizations** page, click a username in the **Users** section.
4. Click **Reset Password**.

   After the password is reset, a message is displayed, which indicates that the password has been reset. If you want to view the initial password after password reset, click **View Password**.

1.1.8.4.1.9. Disable or enable a user account

You can disable a user account to prevent the user account from logging on to the Apsara Uni-manager Management Console. User accounts that are disabled must be re-enabled before they can be used to log on to the Apsara Uni-manager Management Console again.

Context

By default, user accounts are enabled when they are created.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **System Users** tab.
5. Perform the following operations on the current tab:
   - Select a user account whose **Status** is **Enabled**, choose **More** > **Disable** in the **Actions** column to disable the user account.
   - Select a user whose **Status** is **Disabled**, choose **More** > **Enable** in the **Actions** column to enable the user account.

1.1.8.4.1.10. Delete a user

You can delete a specific user as an administrator.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.

3. On the Enterprise page, use one of the following methods to delete a user:
   - In the left-side navigation pane of the **Enterprise** page, click **Users**. On the page that appears, click the **System Users** tab. Click **More** in the **Actions** column corresponding to the target user, and select **Delete**.
   - In the left-side navigation pane of the **Enterprise** page, click **Organizations**. On the page that appears, find the **Users** section. Find the target user, click **More** in the **Actions** column, and then select **Delete**.
4. Click **OK**.

1.1.8.4.2. Historical users

1.1.8.4.2.1. Query historical users

You can check whether a user has been deleted and restore a user that has been deleted.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **Historical Users** tab.
5. Enter the username that you want to query in the **Username** search box.

   > **Note** You can search for usernames by fuzzy match.

6. Click **Search**.

1.1.8.4.2.2. Restore historical users

An administrator can restore a deleted user account from the **Historical Users** tab.

Context

The basic information such as logon password of a restored user is the same as it was before the user was deleted, except for the organization and role.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **Historical Users** tab.
5. Find the user that you want to restore and click **Restore** in the **Actions** column.
6. In the **Restore User** dialog box, select an organization and a role.
7. Click **OK**.

1.1.8.5. Logon policies

1.1.8.5.1. Create a logon policy

To improve the security of the Apsara Uni-manager Management Console, you can create a logon policy as an administrator to control logon access based on the logon time and user IP address.

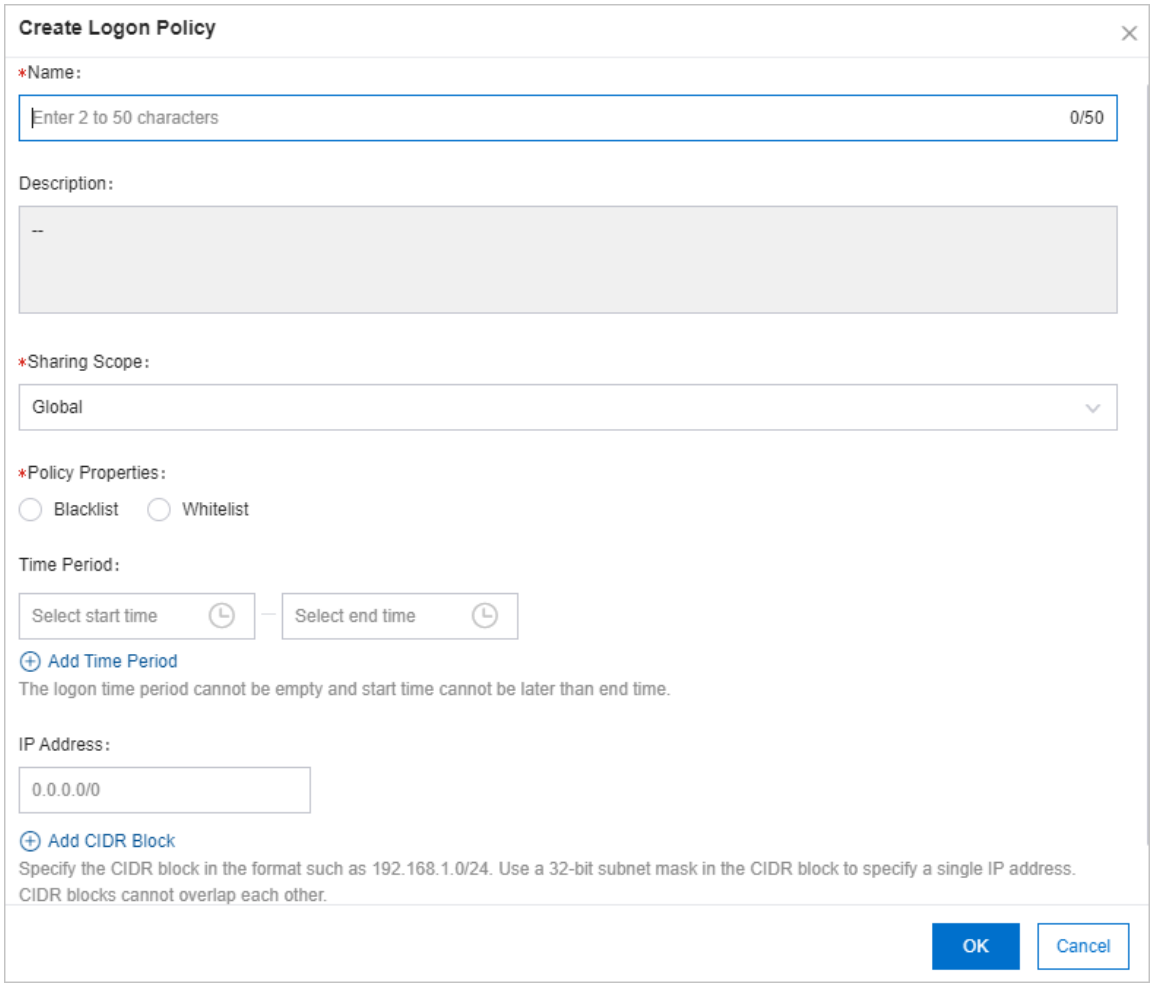
Context

Logon policies are used to control the time period and IP addresses for users to log on. After a user is bound to a logon policy, user logons are restricted based on the logon time and IP addresses specified in the policy.

A default policy without limits on logon time and IP addresses is automatically generated in the Apsara Uni-manager Management Console. The default policy cannot be deleted.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Logon Policies**.
4. In the upper-right corner of the page, click **Create**.
5. In the **Create Logon Policy** dialog box, set Name, Sharing Scope, Policy Properties, Time Period, and IP Address.



Parameters for creating a logon policy

Parameter	Description
Name	The name of the logon policy. The name must be 2 to 50 characters in length and can contain only letters and digits. The name must be unique in the system.
Description	The description of the logon policy.

Parameter	Description
Sharing Scope	The scope in which the role is visible. ○ **Global**: The role is globally visible. The default value is Global. ○ **Current Organization**: The role is visible only in the current organization and is invisible in subordinate organizations. ○ **Subordinate Organization**: The role is visible in the current organization and all its subordinate organizations.
Policy Properties	The authentication method of the logon policy. ○ **Whitelist**: Logon is allowed if the parameter settings are met. ○ **Blacklist**: Logon is denied if the parameter settings are met.
Time Period	The permitted logon time period. When this policy is configured, users can log on to the Apsara Uni-manager Management Console only during the configured period. Specify the time in minutes in a 24-hour clock. Example: `16:32`. **Note** When the Policy Properties parameter is set to Whitelist, you can select No Time Limit.
IP Address	The permitted CIDR block. ○ If the **Policy Properties** parameter is set to **Whitelist**, IP addresses within this CIDR block are allowed to log on to the Apsara Uni-manager Management Console. ○ If the **Policy Properties** parameter is set to **Blacklist**, IP addresses within this CIDR block are not allowed to log on to the Apsara Uni-manager Management Console. **Note** When the Policy Properties parameter is set to Whitelist, you can select No CIDR Block Limit.

1.1.8.5.2. Query a logon policy

You can query the detailed information of a logon policy in the Apsara Uni-manager Management Console.

Context

When the Apsara Uni-manager Management Console provides services, it automatically generates a default policy without limits on the logon time and IP addresses.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Logon Policies**.
4. Enter the name of the policy that you want to view and click **Search**.
5. View the logon policy, including the permitted logon time and IP addresses.

1.1.8.5.3. Modify a logon policy

You can modify the policy name, policy properties, permitted logon time period, and IP addresses of a logon policy.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Logon Policies**.
4. Find the logon policy that you want to modify and choose **More** > **Modify** in the **Actions** column.
5. In the **Modify Logon Policy** dialog box, modify the logon policy information.
6. Click **OK**.

1.1.8.5.4. Disable a logon policy

You can disable logon policies that are no longer needed.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Logon Policies**.
4. Find the logon policy that you want to disable and choose **More** > **Disable** in the **Actions** column.

1.1.8.5.5. Enable a logon policy

You can re-enable disabled logon policies.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Logon Policies**.
4. Click **More** in the **Actions** column corresponding to a policy, and choose **Enable** from the shortcut menu.

1.1.8.5.6. Delete a logon policy

You can delete logon policies that are no longer needed.

Prerequisites

The logon policy to be deleted is not bound to any users. If a logon policy is bound to a user, the logon policy cannot be deleted.

Context

> **Note** The default policy cannot be deleted.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Logon Policies**.
4. Click **More** in the **Actions** column corresponding to a policy, and choose **Delete** from the shortcut menu.
5. In the message that appears, click **OK**.

1.1.8.6. User groups

1.1.8.6.1. Create a user group

You can create a user group in a selected organization and grant batch authorizations to users in the group.

Prerequisites

Before creating a user group, you must create an organization. For more information, see Create an organization.

Context

Relationship between user groups and users:

- A user group can contain zero or more users.
- You can add users to user groups as needed.
- You can add a user to multiple user groups.

Relationship between user groups and organizations:

- A user group can only belong to a single organization.
- You can create multiple user groups in an organization.

Relationship between user groups and roles:

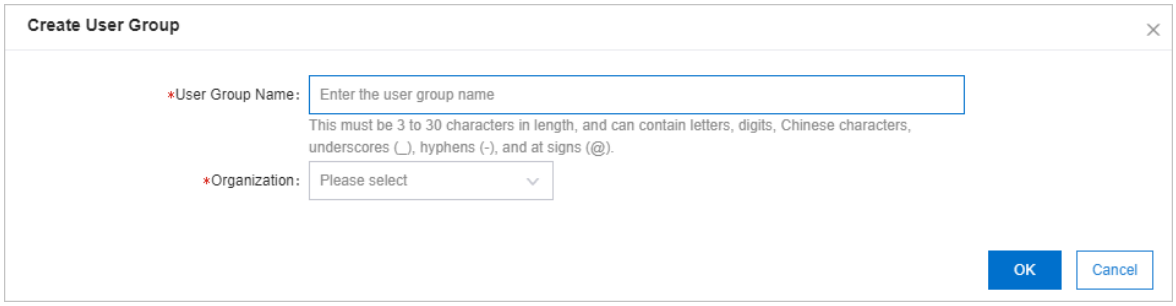
- A user group can only be bound to a single role.
- A role can be associated with multiple user groups.
- When a role is associated with a user group, the role permissions are automatically granted to users in the user group.

Relationship between user groups and resource sets:

- You can add zero or more user groups to a resource set.
- A user group can be added to multiple resource sets.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **User Groups**.
4. In the upper-right corner of the page, click **Create User Group**.
5. In the dialog box that appears, set **User Group Name** and **Organization**.



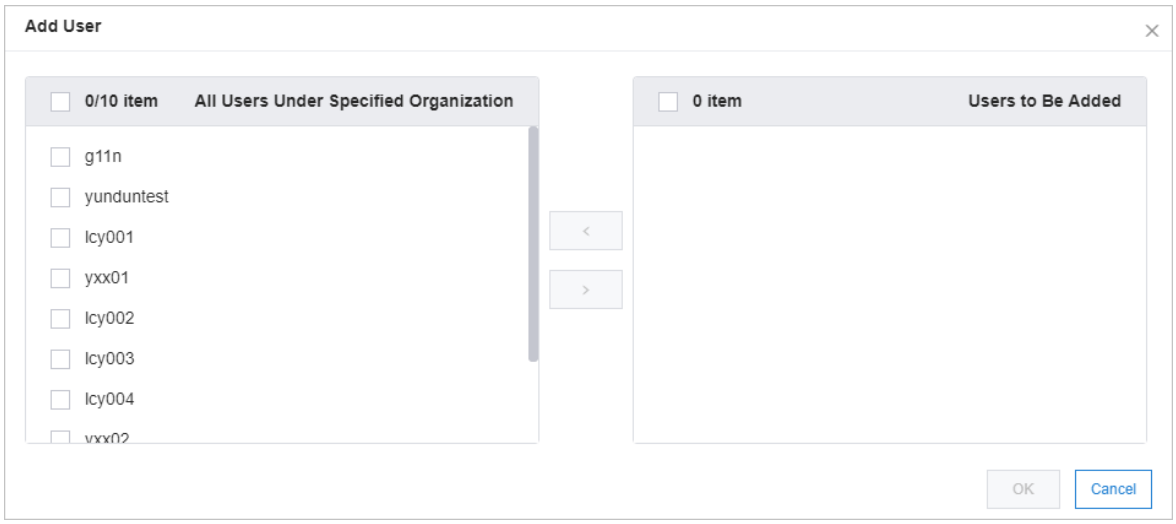
6. Click **OK**.

1.1.8.6.2. Add users to a user group

You can add users to a user group.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **User Groups**.
4. Click **Add User** in the **Actions** column corresponding to a user group.
5. Select the names of users to be added from the left list, and click the right arrow to move them to the right list.



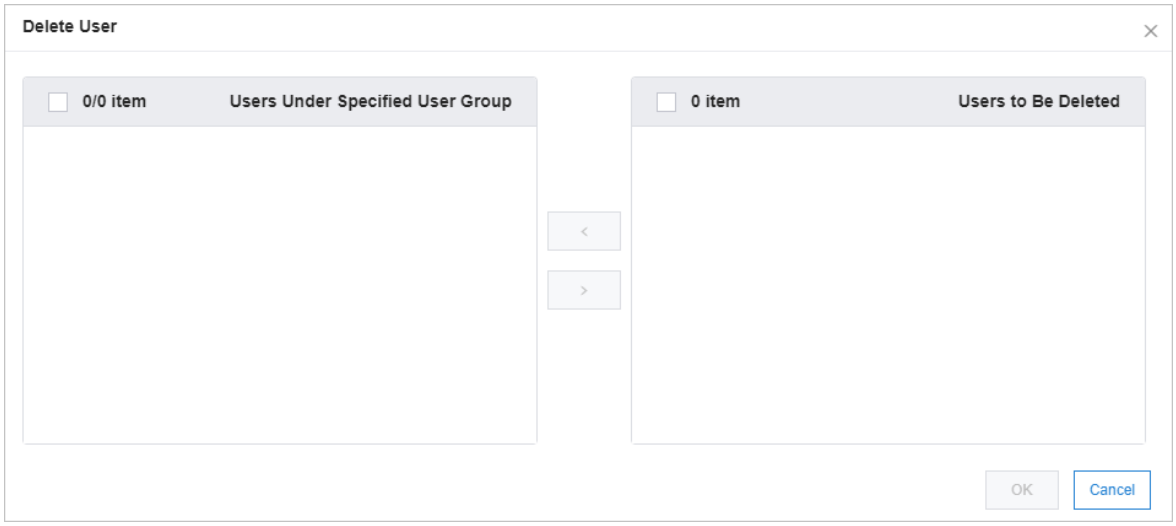
6. Click **OK**.

1.1.8.6.3. Delete users from a user group

You can delete users from a user group.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **User Groups**.
4. Click **Delete User** in the **Actions** column corresponding to a user group.
5. Select the names of users to be deleted from the **Users Under Specified User Group** list, and click the right arrow to move them to the **Users to Be Deleted** list.



6. Click **OK**.

1.1.8.6.4. Add a role

You can add a role to a user group and assign the role to all users in the group.

Context

> **Note** You can add only one role to a user group.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **User Groups**.
4. Click **Add Role** in the **Actions** column corresponding to a user group.
5. In the dialog box that appears, select a role.
6. Click **OK**.

1.1.8.6.5. Delete a role

You can delete existing roles.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **User Groups**.
4. Find the user group from which you want to delete a role and click **Delete Role** in the **Actions** column.
5. In the **Confirm** message, click **OK**.

1.1.8.6.6. Modify the name of a user group

You can modify the names of user groups.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **User Groups**.
4. Click **Edit User Group** in the **Actions** column corresponding to a user group.
5. In the dialog box that appears, enter the new name.
6. Click **OK**.

1.1.8.6.7. Delete a user group

You can delete user groups that are no longer needed.

Prerequisites

The user group to be deleted is unbound from all roles. If a user group is bound to a role, the user group cannot be deleted.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **User Groups**.
4. Find the user group that you want to delete and click **Delete User Group** in the **Actions** column.
5. In the **Confirm** message, click **OK**.

1.1.8.7. Resource pools

1.1.8.7.1. Update associations

You can deploy the Apsara Uni-manager Management Console in multiple regions. You can update the associations between organizations and regions.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Resource Pools**.
4. In the left-side organization navigation tree, click the name of the organization that you want to update.
5. In the corresponding region list, select the names of regions to be associated.
6. Click **Update Association**.

1.1.8.8. Change the ownership of an instance

You can change the ownership of an instance from one resource set to another.

Change the ownership of an instance

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Change Ownership**.
4. Click the ▸ icon to the left of an organization and click a resource set.

5. In the resource list on the right side of the page, set a service type and a resource type, enter an instance ID, and then click **Search** to query the instance.
6. Click **Change ownership** in the **Actions** column corresponding to the instance to change the ownership of the instance to another resource set.
7. Click **Change sharing scope** in the **Actions** column corresponding to the instance to change the sharing scope of the instance.
   - Current Organization and Subordinate Organizations: The instance can be shared by the organization that contains the resource set to which the instance belongs and by subordinate organizations.
   - Current Resource Set: The instance can be shared by the resource set to which the instance belongs.
   - Current Organization: The instance can be shared by the organization that contains the resource set to which the instance belongs.
8. In the **Change Resource Set** dialog box, select a resource set and click **OK**.

1.1.8.9. Cloud instances

1.1.8.9.1. Manage Apsara Stack cloud instances

1.1.8.9.1.1. Export data of the current cloud

You can export the data of secondary Apsara Stack nodes to a configuration file. This can be used by the primary node to manage nodes in a centralized manner.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**. In the left-side navigation pane of the Enterprise page, click **Cloud Instances**.
3. Click the **Apsara Stack Management** tab.
4. Click **Collect Data of Current Cloud** to collect the deployment information of the current cloud.
5. Click **Export** to export the information in the JSON format.

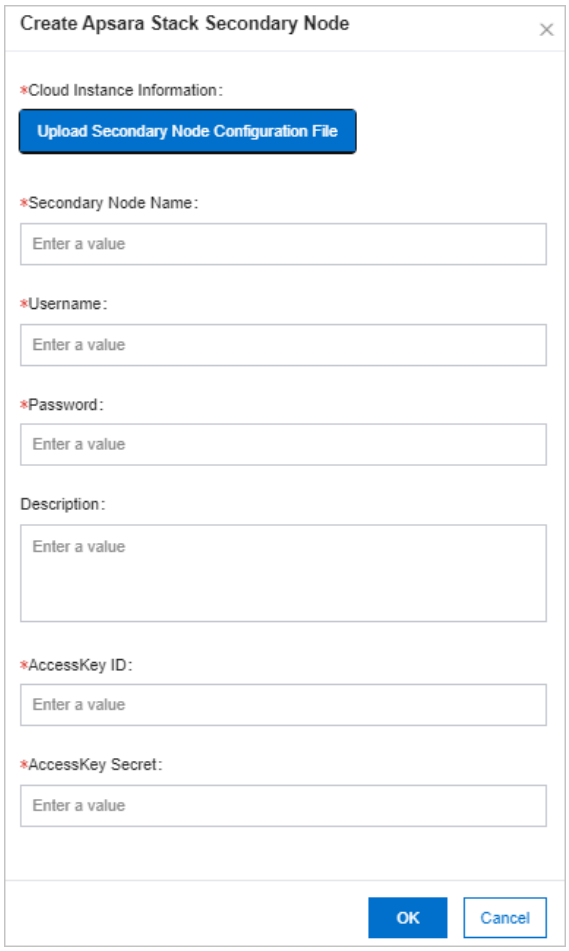
1.1.8.9.1.2. Add a secondary Apsara Stack node

You can add the configuration information of secondary Apsara Stack nodes to the multi-cloud configuration of the primary Apsara Stack node for centralized management.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**. In the left-side navigation pane of the Enterprise page, click **Cloud Instances**.
3. Click the **Apsara Stack Management** tab.
4. Click **Import**.

5. In the **Create Apsara Stack Secondary Node** dialog box, enter the configuration information of a secondary node and click **OK**.



Parameter	Description
Cloud Instance Information	The configuration file of the secondary node. For more information, see Export data of the current cloud.
Secondary Node Name	The name of the secondary node.
Username	The username of the operations administrator that manages the secondary node.
Password	The password of the operations administrator that manages the secondary node.
Description	The description of the secondary node.

Parameter	Description
AccessKey ID	The AccessKey ID of the operations administrator that manages the secondary node. For more information, see View the AccessKey pair of your Apsara Stack tenant.
AccessKey Secret	The AccessKey secret of the operations administrator that manages the secondary node. For more information, see View the AccessKey pair of your Apsara Stack tenant.

Notice

You must create an operations administrator account in the secondary node. This account is for dedicated use by the primary node and cannot be the default operations administrator account.

1.1.8.9.1.3. View managed cloud instances

You can use the multi-cloud management feature to view the details of all managed cloud instances.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**. In the left-side navigation pane of the Enterprise page, click **Cloud Instances**.
3. Click the **Apsara Stack Management** tab.

   You can view the name, description, cloud type, cloud role, and address of all managed cloud instances.
4. Enter a cloud instance name in the search box and click **Search** to search for the cloud instance.
5. Click **View Details** in the **Actions** column corresponding to the cloud instance.

   In the Manage Cloud Instance message, you can view the version, ASAPI address, and region of the cloud.

1.1.8.9.1.4. Modify a cloud instance

If you want to change the information of a cloud instance for more efficient management, you can modify it in the Apsara Uni-manager Management Console.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**. In the left-side navigation pane of the Enterprise page, click **Cloud Instances**.
3. Click the **Apsara Stack Management** tab.
4. Enter the name of a cloud instance that you want to modify in the search box and click **Search** to search for the cloud instance.
5. Click **Edit** in the **Actions** column corresponding to the cloud instance.

6. In the **Edit Cloud Instance** dialog box, set **Cloud Name**, **Username**, **Password**, **Description**, **AccessKey ID**, **AccessKey Secret**, **Longitude**, and **Latitude**, and click **OK**.

1.1.8.9.1.5. Manage cloud instances

You can manage Apsara Stack cloud instances to check whether they can be connected.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**. In the left-side navigation pane of the Enterprise page, click **Cloud Instances**.
3. Click the **Apsara Stack Management** tab.
4. Enter a cloud instance name in the search box and click **Search** to search for the cloud instance.
5. Click **Manage** in the **Actions** column corresponding to the cloud instance.
6. In the **Manage Cloud Instance** dialog box, click **Test Connectivity**.

1.1.8.9.2. Manage VMware nodes

1.1.8.9.2.1. Add a VMware node

You can add the configuration information of VMware nodes to the Apsara Stack VMware management configuration for centralized management.

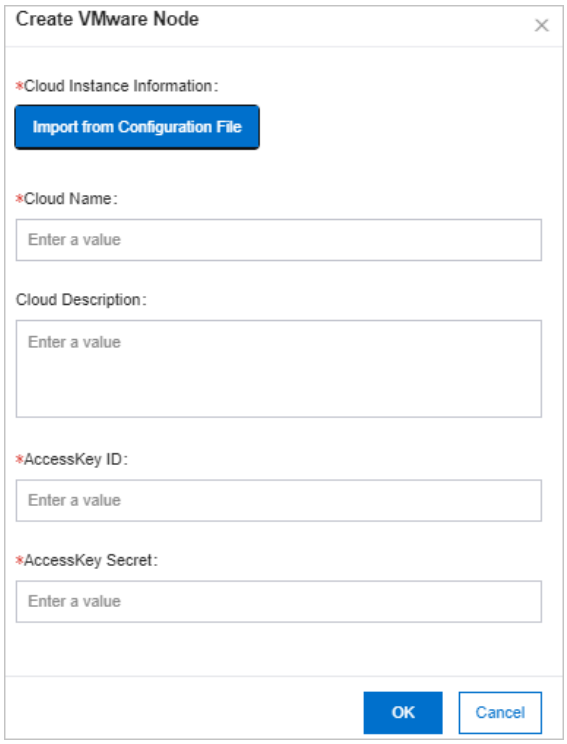
Prerequisites

- The configuration file of a VMware node is obtained from the deployment personnel.
- The VMware node is configured.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the Enterprise page, click **Cloud Instances**.
4. Click the **VMware Management** tab.
5. Click **Create VMware Node**.

6. In the **Create VMware Node** dialog box, enter the configuration information of a VMware node and click **OK**.



Parameter	Description
Cloud Instance Information	The configuration file of the VMware node.
Cloud Name	The name of the VMware node.
Cloud Description	The description of the VMware node.
AccessKey ID	The AccessKey ID in the configuration file of the VMware node.
AccessKey Secret	The AccessKey secret in the configuration file of the VMware node.

1.1.8.9.2.2. Modify a VMware node

If you want to change the information of a VMware node for more efficient management, you can modify it in the Apsara Uni-manager Management Console.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**. In the left-side navigation pane of the Enterprise page, click **Cloud Instances**.

3. Click the **VMware Management** tab.
4. Enter the name of a VMware node that you want to modify in the search box and click **Search** to search for the VMware node.
5. Click **Edit** in the **Actions** column corresponding to the VMware node.
6. In the **Edit Cloud Instance** dialog box, set **Cloud Name**, **Cloud Description**, **AccessKey ID**, and **AccessKey Secret**, and click **OK**.

1.1.8.9.2.3. Test VMware node connectivity

You can manage VMware nodes to check whether they can be connected.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**. In the left-side navigation pane of the Enterprise page, click **Cloud Instances**.
3. Click the **VMware Management** tab.
4. Enter a VMware node name in the search box and click **Search** to search for the VMware node.
5. Click **Manage** in the **Actions** column corresponding to the VMware node.
6. In the **Manage Cloud Instance** dialog box, click **Test Connectivity**.

1.1.8.10. Data permissions

1.1.8.10.1. Overview

Data permission management allows you to specify which users can access instances of a specific service, grant data access permissions to the users, and view and modify the data permissions in all the RAM policies attached to specified users.

Apsara Stack controls users and permissions by managing their visibility and operability in the Apsara Uni-manager Management Console. Many Apsara Stack cloud services are directly used by calling their API operations or SDKs instead of in the console. In this case, data access permissions must be controlled by RAM permission verification provided by the cloud services.

RAM policies are configured for such cloud service instances for access control. Automatic judgment is used when personnel are added to or removed from resource sets. However, this judgement method can affect performance and has a high error rate in complex scenarios. To solve this problem, the authorization of cloud services that require data access permissions is separately managed. Organization administrators can configure the data permissions granted to related personnel on the data authorization page.

1.1.8.10.2. Set the data permissions of resource instances

Organization administrators can set the data permissions of resource instances to allow or prohibit access to and operations on cloud services in the Apsara Uni-manager Management Console.

Prerequisites

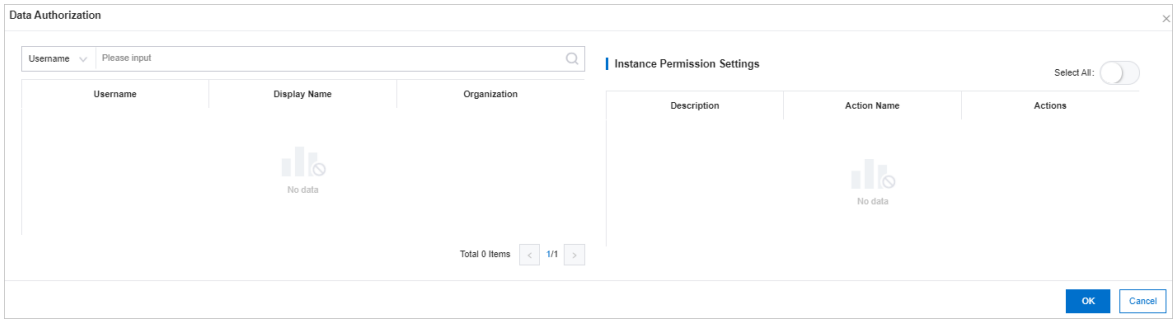
The cloud services that support data authorization include Message Queue (MQ), Object Storage Service (OSS), Log Service, DataHub, and Container Service.

Procedure

1. Log on to the Apsara Uni-manager Management Console.

2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the Enterprise page, click **Data Permissions**.
4. Click a resource set and click a product type on the right side of the page.
5. Click **Authorize** in the **Actions** column corresponding to the instance that you want to manage.
6. In the Data Authorization dialog box, select a user on the left side.
7. Turn on or off the data permission switches in the Actions column on the right side.

   You can also turn on or off the Select All switch to manage permissions in batches.



8. Click **OK**.

1.1.8.10.3. Edit user permissions

You can use JSON statements to edit user permissions.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the Enterprise page, click **Data Permissions**.
4. In the organization navigation tree, click the ▸ icon to the left of the organization that contains the user you want to manage.
5. Click **Users**.
6. Enter the username in the search box and click **Search**.
7. Click **Edit Permissions** in the **Actions** column corresponding to the user.
8. In the Edit Permissions dialog box, select a data permission on the left side and click OK.

   If no permissions are available, specify a policy in the text editor. For more information about the syntax and structure of a policy, see Permission policy structure and syntax.

1.1.8.10.4. View the permissions of a user

You can view the existing policies of a user.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the Enterprise page, click **Data Permissions**.

4. In the organization navigation tree, find the organization that contains the user you want to manage and click the ▸ icon.
5. Click **Users**.
6. Enter the username in the search box and click **Search**.
7. Click **View Permissions** in the **Actions** column corresponding to the user.

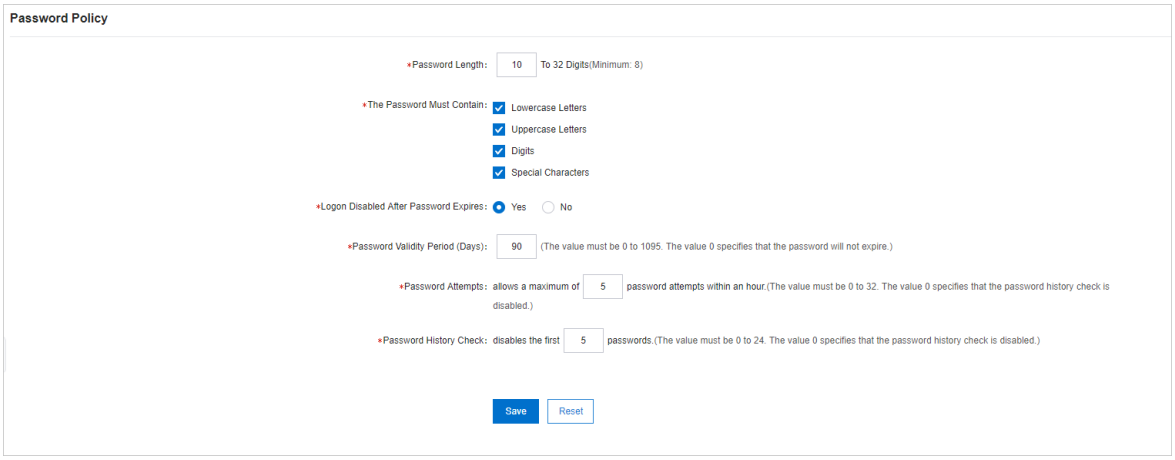
1.1.9. Configurations

1.1.9.1. Password policies

You can configure password policies for user logons.

Procedure

1. Log on to the Apsara Uni-manager Management Console as a platform administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Security Policies**.
4. On the **Password Policy** tab, set the password policy parameters.



To restore to the default password policy, click **Reset**.

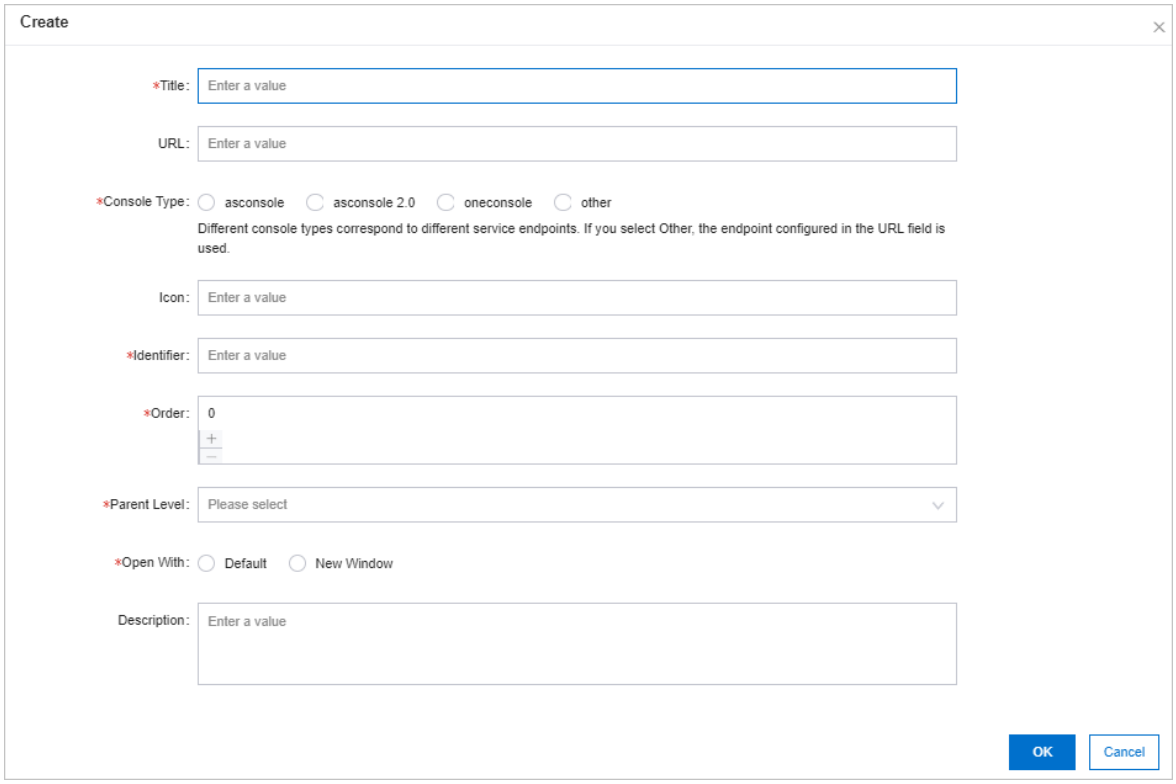
1.1.9.2. Menus

1.1.9.2.1. Create a menu

You can create a menu and add its URL to the Apsara Uni-manager Management Console for quick access.

Procedure

1. Log on to the Apsara Uni-manager Management Console as a platform administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Menu Settings**.
4. On the **Main Menu** page, click **Create** in the upper-right corner.
5. In the **Create** dialog box, set the menu parameters.



Menu parameters

Parameter	Description
Title	The display name of the menu.
URL	The URL of the menu.
Console Type	Different console types correspond to different domain names. ○ **oneconsole**: You need only to enter the path in the URL field. The domain name is automatically matched. ○ **asconsole**: You need only to enter the path in the URL field. The domain name is automatically matched. ○ **other**: You must enter the domain name in the URL field.
Icon	The icon displayed in the left-side navigation pane. The icon cannot be changed.
Identifier	The unique identifier of the menu in the system. This identifier can be used to indicate whether the menu is selected in the navigation bar. The identifier cannot be changed.
Order	The display order among the same-level menus. The larger the value, the lower the display order. Leave the Order field empty.
Parent Level	The displayed tree structure.
Open With	Specifies whether to open the menu in the current window or in a new window.
Description	The description of the menu.

1.1.9.2.2. Modify a menu

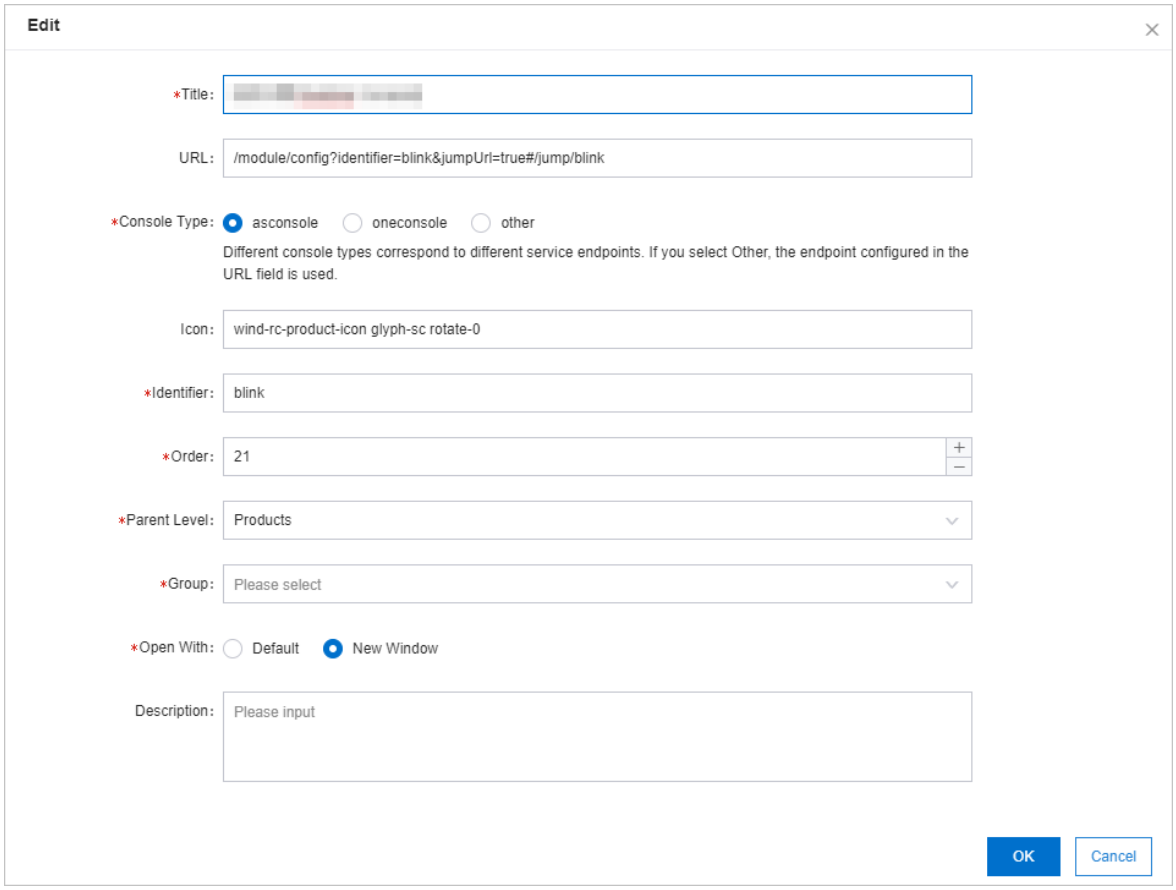
You can modify an existing menu, including the menu name, URL, icon, and menu order.

Prerequisites

Default menus cannot be modified.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Menu Settings**.
4. Click **Edit** in the **Actions** column corresponding to a menu.
5. In the **Edit** dialog box, modify the relevant information of the menu.



1.1.9.2.3. Delete a menu

You can delete menus that are no longer needed.

Prerequisites

Default menus cannot be deleted.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Menu Settings**.
4. Click **Delete** in the **Actions** column corresponding to a menu.

5. In the message that appears, click **OK**.

1.1.9.2.4. Display or hide menus

You can display or hide menus as follows:

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Menu Settings**.
4. Select or clear the check box in the **Displayed** column corresponding to a menu.

1.1.9.3. Specifications

1.1.9.3.1. Specification parameters

This topic describes the specification parameters of each resource type.

OSS

Parameter	Description
Specifications	The specifications that can be configured for Object Storage Service (OSS).
Specifications Description	The description of the specifications that can be configured for OSS.

NAT Gateway

Parameter	Description
Specifications	The specifications that can be configured for NAT Gateway.
Specifications Description	The description of the specifications that can be configured for NAT Gateway.

AnalyticDB for PostgreSQL

Parameter	Description
Specifications	The specifications that can be configured for AnalyticDB for PostgreSQL.
Specifications Name	The name of the specifications that can be configured for AnalyticDB for PostgreSQL.
CPU	The total number of CPU cores that can be configured for AnalyticDB for PostgreSQL.
Memory	The memory size that can be configured for AnalyticDB for PostgreSQL.

Parameter	Description
Storage Space	The total storage size that can be configured for AnalyticDB for PostgreSQL.
Version	The version number of AnalyticDB for PostgreSQL.
Node	The number of nodes that can be configured for AnalyticDB for PostgreSQL.

SLB

Parameter	Description
Specifications	The specifications that can be configured for Server Load Balancer (SLB).
Specifications Name	The name of the specifications that can be configured for SLB.
Maximum Connections	The maximum number of connections that can be configured for SLB.
New Connections	The number of new connections that can be configured for SLB.
QPS	The queries per second (QPS) that can be configured for SLB.
Description	The description of the specifications that can be configured for SLB.

ApsaraDB RDS

Parameter	Description
Engine Type	The engine type that can be configured for ApsaraDB RDS.
Minimum Storage (GB)	The minimum amount of storage space that can be configured for ApsaraDB RDS.
Maximum Storage (GB)	The maximum amount of storage space that can be configured for ApsaraDB RDS.
Specifications Name	The name of the specifications that can be configured for ApsaraDB RDS.
Version	The version number of ApsaraDB RDS.
CPUs	The number of CPU cores that can be configured for ApsaraDB RDS.
Maximum Connections	The maximum number of connections that can be configured for ApsaraDB RDS.
Storage	The amount of storage space that can be configured for ApsaraDB RDS.

Parameter	Description
Memory (GB)	The memory size that can be configured for ApsaraDB RDS.
Share Type	The share type that can be configured for ApsaraDB RDS.

PolarDB-X

Parameter	Description
Instance Type	The instance type that can be configured for PolarDB-X.
Instance Type Name	The name of the instance type that can be configured for PolarDB-X.
Specifications	The specifications that can be configured for PolarDB-X.
Specifications Name	The name of the specifications that can be configured for PolarDB-X.

ECS

Parameter	Description
Instance Family	The instance family that is divided into different instance types based on the scenarios for which they are suitable.
Specifications Level	The level of the specifications that can be configured for Elastic Compute Service (ECS).
vCPUs	The maximum number of vCPUs that can be configured for ECS.
Memory (GB)	The memory size that can be configured for ECS.
Instance Specifications	The instance type that can be configured for ECS.
GPU Type	The GPU type that can be configured for ECS.
GPUs	The number of GPUs that can be configured for ECS.
Supported ENIs	The number of Elastic Network Interfaces (ENIs) that can be configured for ECS.
Number Of Private IP Addresses	The number of private IP addresses that can be configured for ECS.

IPv6 Translation Service

Parameter	Description
Specifications	The specifications that can be configured for IPv6 Translation Service.
Specifications Name	The name of the specifications that can be configured for IPv6 Translation Service.

KVStore for Redis

Parameter	Description
Specifications Name	The name of the specifications that can be configured for KVStore for Redis.
Instance Specifications	The instance type that can be configured for KVStore for Redis.
Maximum Connections	The maximum number of connections that can be configured for KVStore for Redis.
Maximum Bandwidth	The maximum bandwidth that can be configured for KVStore for Redis.
CPUs	The number of CPU cores that can be configured for KVStore for Redis.
Version	The version number of KVStore for Redis.
Architecture	The architecture of KVStore for Redis.
Node Type	The node type of KVStore for Redis.
Service Plan	The service plan that can be configured for KVStore for Redis.

ApsaraDB for MongoDB

Parameter	Description
Specifications	The specifications that can be configured for ApsaraDB for MongoDB.
Specifications Name	The name of the specifications that can be configured for ApsaraDB for MongoDB.
Engine Type	The engine type that can be configured for ApsaraDB for MongoDB.
Version	The version number of ApsaraDB for MongoDB.
Serial Number	The serial number of ApsaraDB for MongoDB.
Sequence Description	The description of the serial number of ApsaraDB for MongoDB.
Maximum Connections	The maximum number of connections that can be configured for ApsaraDB for MongoDB.
IOPS	The input/output operations per second (IOPS) of ApsaraDB for MongoDB.
Storage Space	The amount of storage space that can be configured for ApsaraDB for MongoDB.
Minimum Storage	The minimum amount of storage space that can be configured for ApsaraDB for MongoDB.

Parameter	Description
Maximum Storage	The maximum amount of storage space that can be configured for ApsaraDB for MongoDB.

1.1.9.3.2. Create specifications

You can customize specifications for each resource type.

Procedure

1. Log on to the Apsara Uni-manager Management Console as a platform administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Specifications**.
4. Select the resource type for which you want to create specifications and click the **Resource Specifications** tab.
5. On the **Resource Specifications** tab, click **Create Specifications** in the upper-right corner.
6. In the dialog box that appears, set the specifications parameters.

   For more information about specification parameters, see Specification parameters.
7. Click **OK**.

1.1.9.3.3. View specifications

You can view the specifications of each resource type.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Specifications**.
4. Click the resource type for which you want to view specifications.
5. On the **Resource Specifications** tab, set a **region**, **column**, and **value**. The corresponding information is displayed in the specifications list.
6. Click the **Existing Specifications** tab and view the existing specifications and their quantity.

1.1.9.3.4. Disable specifications

By default, the status of newly created specifications is Enabled.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Specifications**.
4. Select the resource type for which you want to disable specifications.
5. Click **Disable** in the **Actions** column corresponding to the target specifications.
6. In the message that appears, click **OK**.

1.1.9.3.5. Export specifications

You can export specifications that you want to view and share.

Procedure

1. Log on to the Apsara Uni-manager Management Console as a platform administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Specifications**.
4. Click the resource type whose specifications are created.
5. In the upper-right corner of the page, click **Export**.
6. Save the specifications file to a path.

1.1.9.3.6. View specifications of each resource type in previous versions

You can view specifications of each resource type in previous versions.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Specifications**.
4. On the Specifications page, click the resource type for which you want to view specifications.
5. Click the **Specifications History** tab. View the detailed information in the specifications list.

1.1.9.4. Message center

1.1.9.4.1. View internal messages

You can view the IDs and creation time of all internal messages, including unread and read messages.

Context

When an instance is created in a resource, all users that have read and operation permissions on this resource will receive the message that the instance is created.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, move the pointer over the icon and click **More**.
3. In the left-side navigation pane of the **Message Center** page, click the target message scope.
   - Choose **Internal Messages > All Messages** to view all messages, including unread and read messages.
   - Choose **Internal Messages > Unread Messages** to view unread messages.
   - Choose **Internal Messages > Read Messages** to view read messages.

1.1.9.4.2. Mark messages as read

You can mark unread messages as read messages to facilitate message management.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, move the pointer over the icon and click **More**.
3. In the left-side navigation pane of the **Message Center** page, choose **Internal Messages > Unread Messages**.

   In the upper part of the **Unread Messages** page, click different message types to filter messages.
4. On the Unread Messages page, find the message that you want to mark as read and click **Mark as Read** in the **Actions** column.

   You can also select the check boxes to the left of messages and click **Batch Read** in the lower-left corner of the page.
5. In the **Mark as Read** message, click **OK**.

1.1.9.4.3. Delete a message

You can delete messages that are no longer needed.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, move the pointer over the icon and click **More**.
3. In the left-side navigation pane of the **Message Center** page, choose **Internal Messages > Unread Messages**.
4. Find the message that you want to delete on the All Messages tab or other tabs and click **Delete**.

   You can also select the check box to the left of the **ID** of the message that you want to delete and click **Batch Delete** in the lower-left corner of the page.

1.1.9.5. Resource pool management

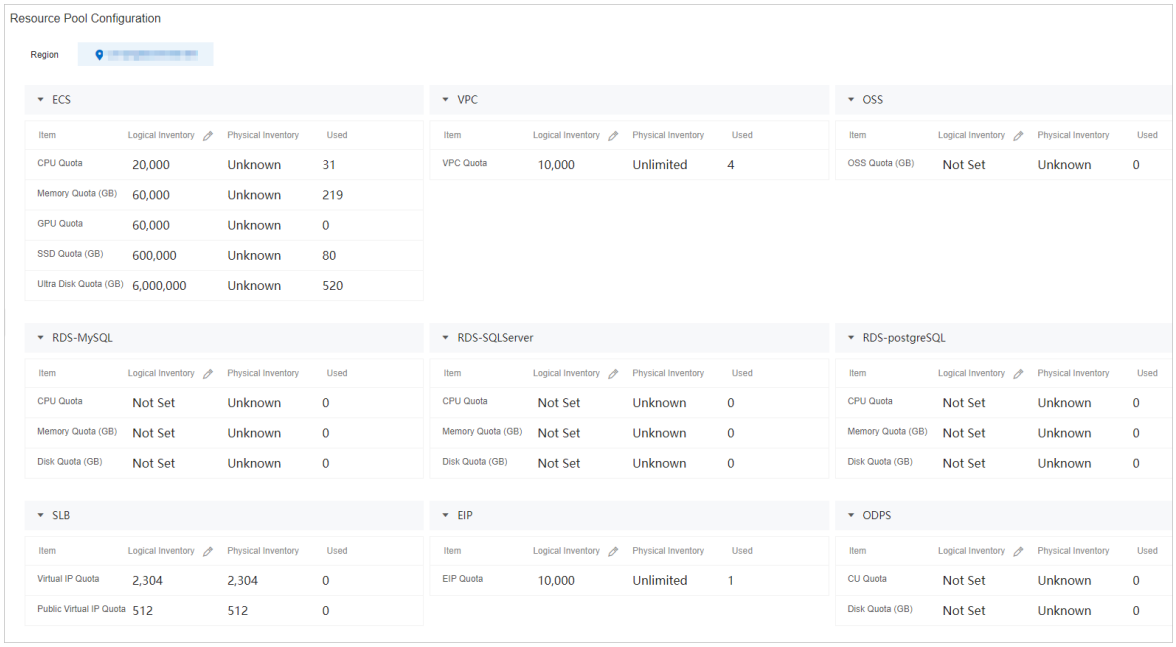
You can modify the maximum usage of each resource.

Prerequisites

- If the physical inventory is unlimited, the logical inventory cannot be less than the used inventory.
- If the physical inventory is limited, the logical inventory cannot be less than the used inventory or greater than the physical inventory.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Resource Pool Management**.
4. On the **Resource Pools** page, click the icon in the module that you want to modify and modify the number of resources.



5. Click the icon to complete modification.

1.1.10. Operations

1.1.10.1. Quotas

1.1.10.1.1. Quota parameters

This topic describes the quota parameters of each service.

An organization administrator can set resource quotas and create resources within the allowed quotas for the organization. When the quotas for the organization are used up, the system does not allow the organization administrator to create more resources for the organization. To create more resources, you must first increase the quotas for the organization.

If no quotas are set, you can create an unlimited amount of resources.

ECS

Parameter	Description
CPU Quota (Cores)	The total number of CPU cores that you can configure for Elastic Compute Service (ECS) and the number of used cores.
Memory Quota (GB)	The total memory size that you can configure for ECS.
GPU Quota (Cores)	The total number of GPU cores that you can configure for ECS.
SSD Quota (GB)	The total SSD capacity that you can configure for ECS.
Ultra Disk Quota (GB)	The total number of disks that you can configure for an ECS instance.

VPC

Parameter	Description
VPC Quota	The maximum number of virtual private clouds (VPCs) that you can configure.

OSS

Parameter	Description
OSS Quota (GB)	The maximum capacity that you can allocate for Object Storage Service (OSS).

RDS-MySQL

Parameter	Description
CPU Quota (Cores)	The total number of CPU cores that you can configure for ApsaraDB RDS for MySQL and the number of used cores.
Memory Quota (GB)	The total memory size that you can configure for ApsaraDB RDS for MySQL.
Disk Quota (GB)	The total storage size that you can configure for ApsaraDB RDS for MySQL.

RDS-PolarDB

Parameter	Description
CPU Quota (Cores)	The total number of CPU cores that you can configure for PolarDB and the number of used cores.
Memory Quota (GB)	The total memory size that you can configure for PolarDB.
Disk Quota (GB)	The total storage size that you can configure for PolarDB.

RDS-SQLServer

Parameter	Description
CPU Quota (Cores)	The total number of CPU cores that you can configure for ApsaraDB RDS for SQL Server and the number of used cores.
Memory Quota (GB)	The total memory size that you can configure for ApsaraDB RDS for SQL Server.
Disk Quota (GB)	The total storage size that you can configure for ApsaraDB RDS for SQL Server.

RDS-PostgreSQL

Parameter	Description
CPU Quota (Cores)	The total number of CPU cores that you can configure for ApsaraDB RDS for PostgreSQL and the number of used cores.

Parameter	Description
Memory Quota (GB)	The total memory size that you can configure for ApsaraDB RDS for PostgreSQL.
Disk Quota (GB)	The total storage size that you can configure for ApsaraDB RDS for PostgreSQL.

SLB

Parameter	Description
Virtual IP Quota (a)	The maximum number of internal IP addresses that you can configure for Server Load Balancer (SLB).
Public Virtual IP Quota	The maximum number of public IP addresses that you can configure for SLB.

EIP

Parameter	Description
EIP Quota	The maximum number of elastic IP addresses (EIPs) that you can configure.

MaxCompute

Parameter	Description
CU Quota (a)	The total number of capacity units (CUs) that you can configure for MaxCompute.
Disk Quota (GB)	The total storage size that you can configure for MaxCompute.

Redis

Parameter	Description
Memory Quota (GB)	The total memory size that you can configure for KVStore for Redis.

DRDS

Parameter	Description
CPU Quota (Cores)	The total number of CPUs that you can configure for PolarDB-X.

NAS

Parameter	Description
Disk Quota (TB)	The total storage size that you can configure for Apsara File Storage NAS.

GPDB

Parameter	Description
CPU Quota (Cores)	The total number of CPU cores that you can configure for AnalyticDB for PostgreSQL and the number of used cores.
Memory Quota (GB)	The total memory size that you can configure for AnalyticDB for PostgreSQL.
Disk Quota (GB)	The total storage size that you can configure for AnalyticDB for PostgreSQL.

ADB

Parameter	Description
CPU Quota (Cores)	The total number of CPU cores that you can configure for AnalyticDB for MySQL V3.0 and the number of used cores.
Memory Quota (GB)	The total memory size that you can configure for AnalyticDB for MySQL V3.0.
Disk Quota (GB)	The total storage size that you can configure for AnalyticDB for MySQL V3.0.

MongoDB

Parameter	Description
CPU Quota (Cores)	The total number of CPU cores that you can configure for ApsaraDB for MongoDB and the number of used cores.
Memory Quota (GB)	The total memory size that you can configure for ApsaraDB for MongoDB.
Disk Quota (GB)	The total storage size that you can configure for ApsaraDB for MongoDB.

AnalyticDB for MySQL V2.0

Parameter	Description
CPU Quota (Cores)	The total number of CPU cores that you can configure for AnalyticDB for MySQL V2.0 and the number of used cores.
Memory Quota (GB)	The total memory size that you can configure for AnalyticDB MySQL V2.0.

1.1.10.1.2. Set quotas for a cloud service

The Apsara Uni-manager Management Console allows you to set quotas to properly allocate resources among organizations.

Prerequisites

You must set quotas for a parent organization before you can set quotas for its subordinate organizations.

Context

If the parent organization has quotas (except when the parent organization is a level-1 organization), the available quotas for a subordinate organization are equal to the quotas for the parent organization minus the quotas for other subordinate organizations.

This topic describes how to modify quotas for Elastic Compute Service (ECS). You can modify quotas for other cloud resources in a similar manner.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Operations**.
3. In the left-side navigation pane of the **Operations** page, click **Quotas**.
4. In the left-side navigation tree, click the name of the organization for which you want to create cloud resources.
5. Select the cloud service for which you want to set quotas. In this example, **ECS** is selected.
6. In the upper-right corner of the quota section, click **Set**.
7. Set the total quotas and click **Save**.

   For more information about quota parameters, see Quota parameters.

1.1.10.1.3. Modify quotas

Administrators can adjust quotas for cloud resources based on organizational requirements.

Context

This topic describes how to modify quotas for ECS. You can modify quotas for other cloud resources in a similar manner.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Operations**.
3. In the left-side navigation pane of the **Operations** page, click **Quotas**.
4. In the left-side navigation tree, click the name of the organization for which you want to create cloud resources.
5. Select the Apsara Stack service for which you want to modify quotas. For this example, **ECS** is selected.
6. In the upper-right corner of the quota area, click **Modify**.
7. Set the total quotas and click **Save**.

   For more information about quota parameters, see Quota parameters.

1.1.10.1.4. Reset quotas

Administrators can reset quotas as needed.

Prerequisites

Before deleting a quota for an organization, make sure that no subordinate organizations have any quotas.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Operations**.
3. In the left-side navigation pane of the **Operations** page, click **Quotas**.

4. In the left-side organization navigation tree, click the name of the target organization.
5. Select the cloud service for which you want to reset quotas. For this example, **ECS** is selected.
6. In the upper-right corner of the quota section, click **Reset**.
7. In the message that appears, click **OK**.

1.1.10.2. Usage statistics

1.1.10.2.1. View the usage statistics of cloud resources

The Apsara Uni-manager Management Console displays statistics about the number of resource instances that run in the Apsara Stack environment by time, organization, resource set, or region. You can also export statistical reports from the Apsara Uni-manager Management Console.

Context

The cloud resources that can be metered include Elastic Compute Service (ECS), Virtual Private Cloud (VPC), Server Load Balancer (SLB), Object Storage Service (OSS), ApsaraDB RDS for MySQL, Elastic IP Address (EIP), Apsara File Storage NAS, Tablestore, PolarDB-X, KVStore for Redis, AnalyticDB for MySQL, AnalyticDB for PostgreSQL, ApsaraDB for MongoDB, Message Queue (MQ), ApsaraDB RDS for PostgreSQL, ApsaraDB RDS for SQL Server, Log Service, ECS disks, scaling group rules, ApsaraDB for HBase, API gateways, Key Management Service (KMS), AnalyticDB for MySQL V2.0, and Time Series Database (TSDB).

This topic describes how to modify quotas for ECS. You can set quotas for other cloud resources in a similar manner.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Operations**.
3. In the left-side navigation pane of the **Operations** page, click **Usage Statistics**.
4. In the **Resource Type** section, click **Elastic Compute Service ECS**.
5. In the **Search Conditions** section, set **Time Period**, **Organization**, **Resource Set**, **Region**, and **Instance ID** to filter resources.

   You can view the statistics in the console or click **Export** in the upper-right corner to export the statistics to your local computer in the XLS format.

   > **Note** In the console, you can view or export up to 1,000 statistical records to an Excel file. Use the statistics query API to obtain more statistical data.

   The exported file is named *<Resource type name>.xls*. Find the downloaded file from the download path of the browser.

1.1.10.3. Statistical analysis

1.1.10.3.1. View reports of current data

You can use reports to view the latest data of each service.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Operations**.

3. In the left-side navigation pane of the Operations page, choose **Statistical Analysis** > **Reports**.
4. Click the tab that you want to view.

   You can click the **Resource Reports**, **Quota Reports**, or **Cloud Monitor Reports** tab.
5. Set **Organizations and Resource Sets** and **Region** and click **Search**.

   You can select the check box to the left of a resource and click **Export Selected Reports** in the lower-left corner to export the report.

1.1.10.3.2. Export reports of current data

You can batch export data that you want to view by cloud service.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Operations**.
3. In the left-side navigation pane of the Operations page, choose **Statistical Analysis** > **Reports**.
4. Click the tab that you want to view.

   You can click the **Resource Reports**, **Quota Reports**, or **Cloud Monitor Reports** tab.
5. Click **Export Reports** on the right side of the page.
6. In the **Select Products to Export** dialog box, select the check box to the left of a service and click **OK**.

   You can also select **Select All** in the lower-left corner and click **OK**.

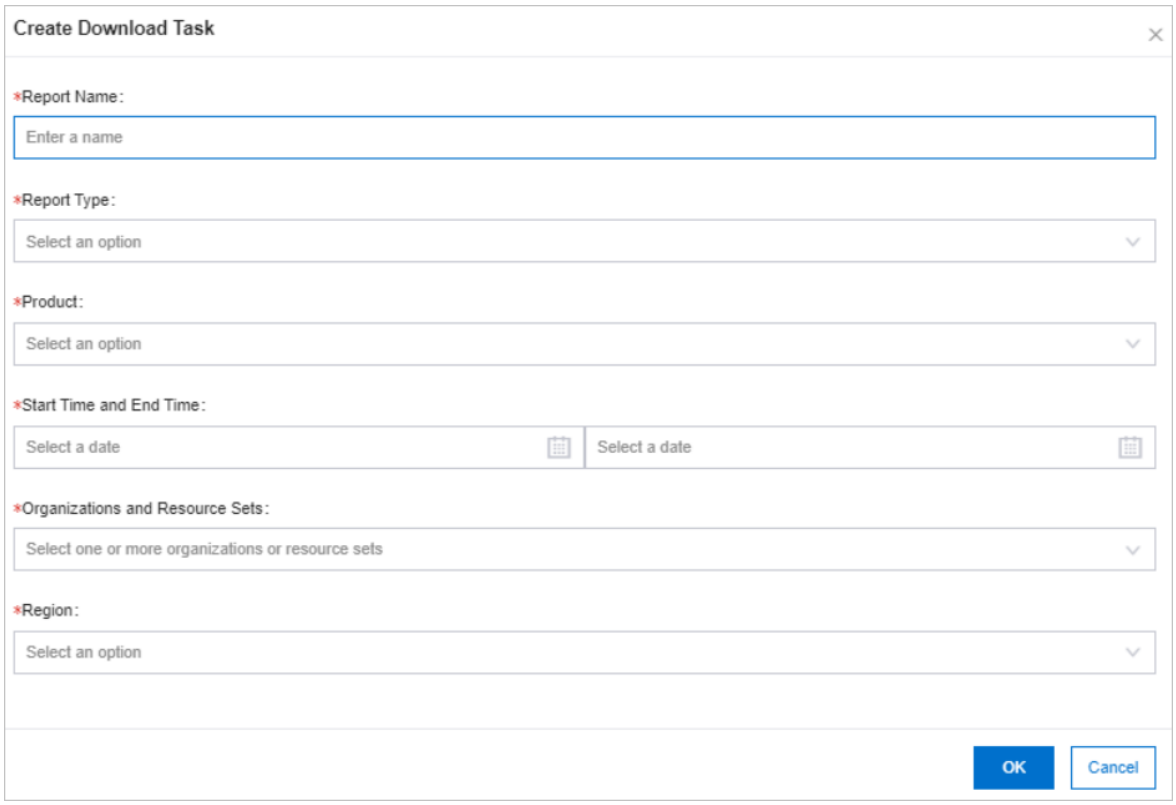
1.1.10.3.3. Download reports of historical data

You can download data reports of cloud services within the specified period of time, resource set, and region by creating download tasks.

Procedure

1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Operations**.
3. In the left-side navigation pane of the Operations page, choose **Statistical Analysis** > **Download Center**.
4. In the upper-right corner of the page, click **Create Download Task**.

5. Enter the information of the download task.



Parameter	Description
Report Name	The name of the report.
Report Type	The type of the report. Valid values: ○ Resource Reports ○ Cloud Monitor Reports
Product	The cloud service for which you want to download reports. You can select multiple cloud services.
Start Time	The start time of the data.
End Time	The end time of the data.
Organizations and Resource Sets	The organization to which the data belongs. You can select multiple organizations.
Region	The region of the data. You can select multiple regions.

6. Click **OK**.

7. After the **Created** message appears, the Download Center page appears. Enter the information of the created report in the search box and click **Search** to search for the created download task.
8. After **In Progress** changes to **Completed** in the **Status** column, click **Download Report** in the Actions column.

1.1.11. Security

1.1.11.1. View operations logs

You can view operations logs to obtain up-to-date information for various resources and functional modules in the Apsara Uni-manager Management Console. You can also export operations logs to your PC.

Procedure

1. Log on to the Apsara Uni-manager Management Console as a security administrator.
2. In the top navigation bar, click **Security**.
3. You can filter logs by username, object, level, source IP address, details, start time, and end time.

   The following table describes the fields in the query result.

   Fields in the query result

Log field	Description
Username	The name of the operator.
Object	The Apsara Stack service on which operations are performed. The operations include creating, modifying, deleting, querying, updating, binding, unbinding, enabling and disabling service instances, applying for and releasing service instances, and changing the ownership of service instances.
Level	The operation level. Valid values: INFO, DEBUG, and ERROR.
Source IP	The IP address of the operator.
Details	A brief introduction of the operation.
Start Time	The time when the operation started.
End Time	The time when the operation ended.

4. (Optional)Click **Export** to export the logs displayed on the current page to your PC in the XLS format.

   The exported log file is named *log.xls* and stored in the *C:\Users\Username\Downloads* directory.

1.1.12. RAM

1.1.12.1. RAM introduction

Resource Access Management (RAM) is a resource access control service provided by Apsara Stack.

You can use RAM to manage users and control which resources are accessible to employees, systems, and applications.

RAM provides the following features:

- RAM role

To authorize a cloud service in a level-1 organization to use other resources in the organization, you must create a RAM role. This role specifies the operations that the cloud service can perform on resources.

Only system administrators and level-1 organization administrators can create RAM roles.

- User group

  You can create multiple users within an organization and grant them different operation permissions on cloud resources.

  You can create RAM user groups to classify and authorize RAM users within your Apsara Stack tenant account. This simplifies the management of RAM users and their permissions.

  You can create RAM permission policies to grant different operation permissions to different user groups.

1.1.12.2. Permission policy structure and syntax

This topic describes the structure and syntax used to create or update permission policies in Resource Access Management (RAM).

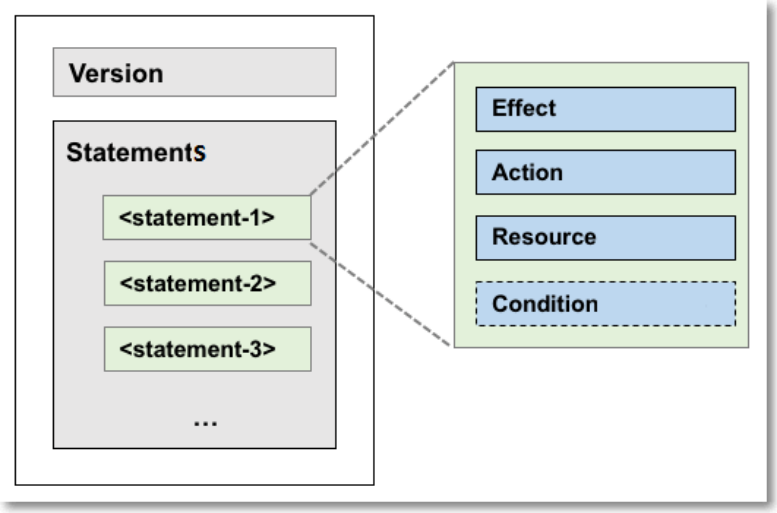
Policy characters and usage rules

- Characters in a policy
  - The following characters are JSON tokens and are included in policies: `{ } [ ] " , :` .
  - The following characters are special characters in the syntax and are not included in policies: `= < > ( ) |` .
- Use of characters
  - If an element can have more than one value, you can perform the following operations:
    - Separate multiple values by using commas (,) as delimiters between each value and use an ellipsis (...) to describe the remaining values. Example: `[ <action_string>, <action_string>, ...]` .
    - Include only one value. Examples: `"Action": [<action_string>]` and `"Action": <action_string>` .
  - A question mark (?) following an element indicates that the element is optional. Example: `<condition_block ? >` .
  - A vertical bar (`|`) between elements indicates multiple options. Example: `("Allow" | "Deny")` .
  - Elements that must be text strings are enclosed in double quotation marks (""). Example: `<version_block> = "Version" : ("1")` .

Policy structure

The policy structure includes the following components:

- The version number.
- A list of statements. Each statement contains the following elements: Effect, Action, Resource, and Condition. The Condition element is optional.



Policy syntax

```
policy  = {
     <version_block>,
     <statement_block>
}
<version_block> = "Version" : ("1")
<statement_block> = "Statement" : [ <statement>, <statement>, ... ]
<statement> = {
    <effect_block>,
    <action_block>,
    <resource_block>,
    <condition_block? >
}
<effect_block> = "Effect" : ("Allow" | "Deny")
<action_block> = ("Action" | "NotAction") :
    ("*" | [<action_string>, <action_string>, ...])
<resource_block> = ("Resource" | "NotResource") :
    ("*" | [<resource_string>, <resource_string>, ...])
<condition_block> = "Condition" : <condition_map>
<condition_map> = {
  <condition_type_string> : {
      <condition_key_string> : <condition_value_list>,
      <condition_key_string> : <condition_value_list>,
      ...
  },
  <condition_type_string> : {
      <condition_key_string> : <condition_value_list>,
      <condition_key_string> : <condition_value_list>,
      ...
  }, ...
}
<condition_value_list> = [<condition_value>, <condition_value>, ...]
<condition_value> = ("String" | "Number" | "Boolean")
```

Description:

- The current policy version is 1.
- The policy can have multiple statements.

- The effect of each statement can be either `Allow` or `Deny`.

  > **Note** In a statement, both the Action and Resource elements can have multiple values.

- Each statement can have its own conditions.

  > **Note** A condition block can contain multiple conditions with different operators and logical combinations of these conditions.

- You can attach multiple policies to a RAM user. If policies that apply to a request include an `Allow` statement and a `Deny` statement, the Deny statement overrides the Allow statement.
- Element value:
  - If an element value is a number or Boolean value, it must be enclosed in double quotation marks ("") in the same way as strings.
  - If an element value is a string, characters such as the asterisk (`*`) and question mark (`?`) can be used for fuzzy matching.
    - The asterisk (`*`) indicates any number (including zero) of allowed characters. For example, `ecs:Describe*` indicates all ECS API operations that start with `Describe`.
    - `The question mark (?)` indicates an allowed character.

Policy format check

Policies are stored in RAM as JSON documents. When you create or update a policy, RAM first checks whether the JSON format is valid.

- For more information about JSON syntax standards, see RFC 7159.
- We recommend that you use tools such as JSON validators and editors to check whether the policies meet JSON syntax standards.

1.1.12.3. RAM roles

1.1.12.3.1. View basic information about a RAM role

You can view basic information about a RAM role, including its user groups and existing permission policies.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. On the Roles page, click the name of the target RAM role.
5. In the basic information section, click the **User Groups** and **Permissions** tabs to view relevant information.

1.1.12.3.2. Create a RAM role

To authorize a cloud service in a level-1 organization to use other resources in the organization, you must create a RAM role. This role contains the operations that the cloud service can perform on resources.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.

3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the upper-right corner of the page, click **Create RAM Role**.
5. On the **Roles - Create RAM Role** page, set **Role Name**, **Description**, and **Sharing Scope**.

   Valid values of the **Sharing Scope** parameter:

   - **Global**

     The role is visible and valid to all organizations involved. The default value is Global.
   - **Current Organization**

     The role is visible and valid to the organization to which the user belongs.
   - **Subordinate Organization**

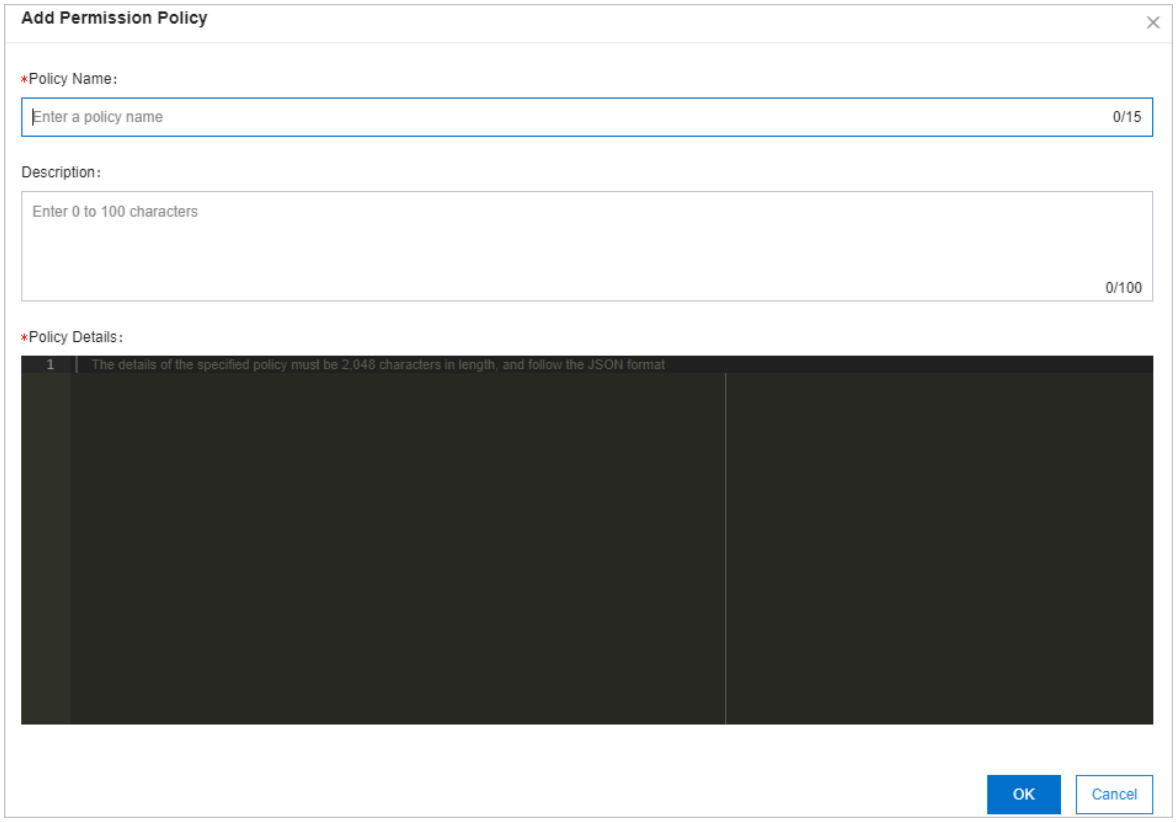
     The role is visible and valid to the organization to which the user belongs and its subordinate organizations.
6. Click **Create**.

1.1.12.3.3. Create a policy

To use a cloud service to access other cloud resources, you must create a policy and attach it to a user group.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, find the RAM role that you want to modify and choose **More** > **Modify** in the **Actions** column to go to the **Roles** page.
5. Click the **Permissions** tab.
6. Click **Add Permission Policy**.
7. In the Add Permission Policy dialog box, enter information of the policy.



For more information about how to enter the policy content, see Permission policy structure and syntax.

1.1.12.3.4. Modify the content of a RAM policy

You can modify the content of a RAM policy.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role list, find the RAM role that you want to modify and choose **More** > **Modify** in the **Actions** column to go to the **Roles** page.
5. Click the **Permissions** tab.
6. Click the name of a policy in the **Permission Policy Name** column.
7. In the **Modify Permission Policy** dialog box, modify the relevant information and click **OK**.

   For more information about how to modify the policy content, see Permission policy structure and syntax.

1.1.12.3.5. Modify the name of a RAM policy

You can modify the name of a RAM policy.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, find the RAM role that you want to modify and choose **More** > **Modify** in the **Actions**

column to go to the **Roles** page.

5. Click the Permissions tab. Click the name of that policy that you want to modify in the **Permission Policy Name** column.
6. In the **Modify Permission Policy** dialog box, modify the policy name.

1.1.12.3.6. Add a RAM role to a user group

You can bind RAM roles to user groups.

Prerequisites

You must create a user group before RAM roles can be added. For more information, see Add a role.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, find the RAM role that you want to modify and choose **More** > **Modify** in the **Actions** column to go to the **Roles** page.
5. Click the **User Groups** tab.
6. Click **Add User Group**. In the Add User Group dialog box, select a user group.
7. Click **OK**.

1.1.12.3.7. Grant permissions to a RAM role

When you grant permissions to a RAM role, all users in the user groups that are assigned this role share the granted permissions.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role list, find the RAM role that you want to modify and choose **More** > **Modify** in the **Actions** column to go to the **Roles** page.
5. Click the **Permissions** tab.
6. Click **Select Existing Permission Policy**.
7. In the dialog box that appears, select a RAM policy and click **OK**.

   If no RAM policies are available, see Add a permission policy.

1.1.12.3.8. Remove permissions from a RAM role

You can remove permissions that are no longer needed from RAM roles.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role list, find the RAM role that you want to modify and choose **More** > **Modify** in the **Actions** column

to go to the **Roles** page.

5. Click the **Permissions** tab.
6. Find the policy that you want to remove and click **Remove** in the **Actions** column.

1.1.12.3.9. Modify a RAM role name

Administrators can modify the names of RAM roles.

Context

The name of a preset role cannot be modified.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, find the RAM role that you want to modify and choose **More** > **Modify** in the **Actions** column to go to the **Roles** page.
5. Move the pointer over the role name and click the icon to enter a new role name.

1.1.12.3.10. Delete a RAM role

This topic describes how to delete a RAM user.

Prerequisites

Before you delete a RAM role, make sure that no policies are attached to the RAM role.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, click **More** in the **Actions** column corresponding to a RAM role, and choose **Delete** from the shortcut menu.
5. In the message that appears, click **OK**.

1.1.12.4. RAM authorization policies

1.1.12.4.1. Create a RAM role

You can create authorization policies and grant them to organizations.

Procedure

1. Log on to the Apsara Uni-manager Management Console as a platform administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **RAM Roles**.
4. In the upper-right corner of the page, click **Create RAM User**.
5. On the **Create RAM User** page, set **Organization** and **Service**.

6. Click **OK**.

1.1.12.4.2. View the details of a RAM role

You can view the details of a RAM role, including its role name, creation time, description, and Alibaba Cloud Resource Name (ARN).

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **RAM Roles**.
4. On the **RAM Users** page, set **Role Name**, **Service Name**, or **Organization Name**, and click **Search** in the upper-right corner.

   To perform another search, click **Clear**.
5. Find the target RAM role and click **Details** in the Actions column.

1.1.12.4.3. View RAM authorization policies

You can view the details of a RAM authorization policy, including its policy name, policy type, default version, description, association time, and policy content.

Prerequisites

A RAM authorization policy is created. For more information, see Create a RAM role.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **RAM Service Linked Role**.
4. On the **RAM Roles** page, set **Role Name** or **Service Name** and click **Search** in the upper-right corner.

   To perform another search, click **Clear**.
5. Find the RAM role that you want to view and click **Details** in the **Actions** column.
6. Click the **Role Policy** tab to view the information of the role authorization policy. Click **Details** in the Actions column to view the policy details.

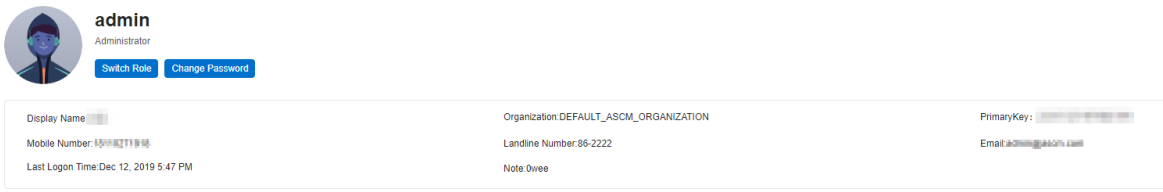
1.1.13. Personal information management

1.1.13.1. Modify personal information

You can modify your personal information to keep it up to date.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **User Information**.



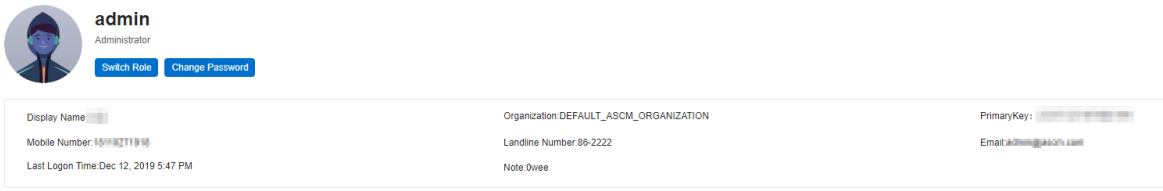
3. Click the icon next to the item that you want to modify.
4. In the Modify User Information dialog box, modify the relevant information.
5. Click **OK**.

1.1.13.2. Change your logon password

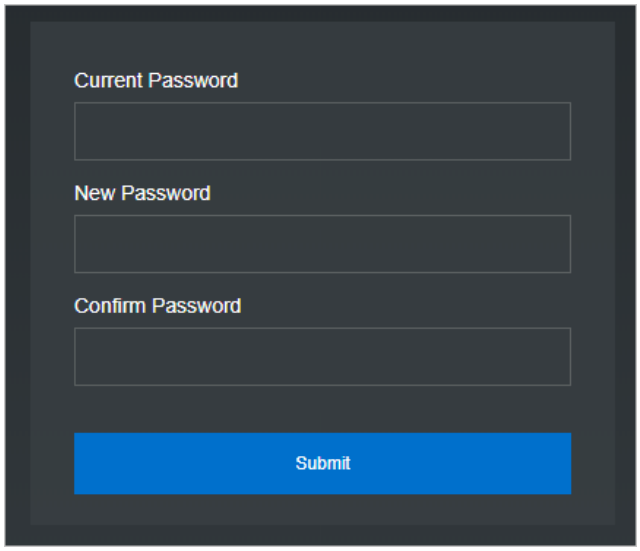
To improve security, you must change your logon password in a timely manner.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the user profile picture and choose **User Information** from the shortcut menu.



3. Click **Change Password**. On the page that appears, set **Current Password**, **New Password**, and **Confirm Password**.



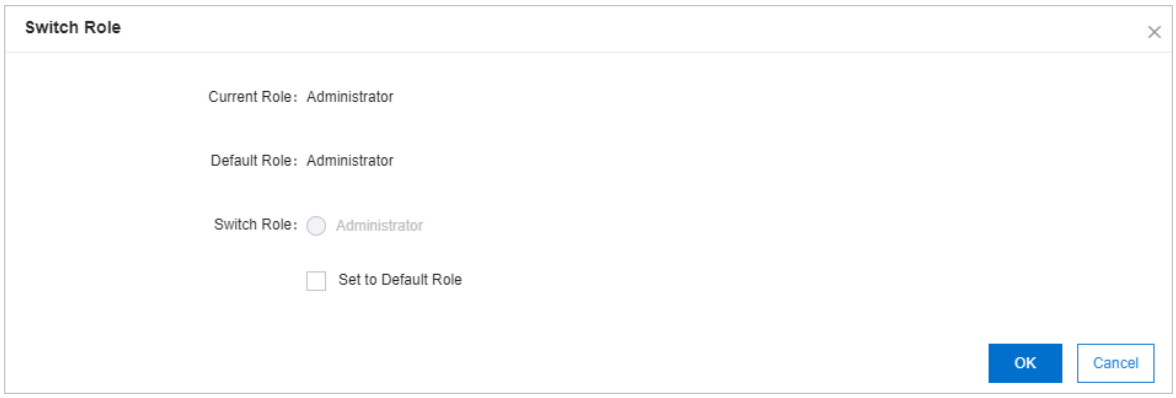
4. Click **Submit**.

1.1.13.3. Switch the current role

You can switch the scope of your current role.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the user profile picture and choose **User Information** from the shortcut menu.
3. Click **Switch Role**.
4. In the **Switch Role** dialog box that appears, select the role that you want to switch to.



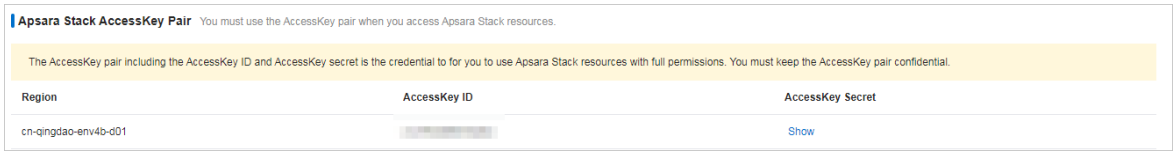
You can also switch back to the default role.

1.1.13.4. View the AccessKey pair of your Apsara Stack tenant account

To secure cloud resources, the system must verify the identity of visitors and ensure that they have the relevant permissions. You must obtain the AccessKey ID and AccessKey secret of your personal account to access cloud resources.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the user profile picture and choose **User Information** from the shortcut menu.
3. In the **Apsara Stack AccessKey Pair** section, view your AccessKey pair.



> **Note** The AccessKey pair is made up of the AccessKey ID and AccessKey secret. These credentials provide you full permissions on Apsara Stack resources. You must keep the AccessKey pair confidential.

2.Elastic Compute Service (ECS)

2.1. User Guide

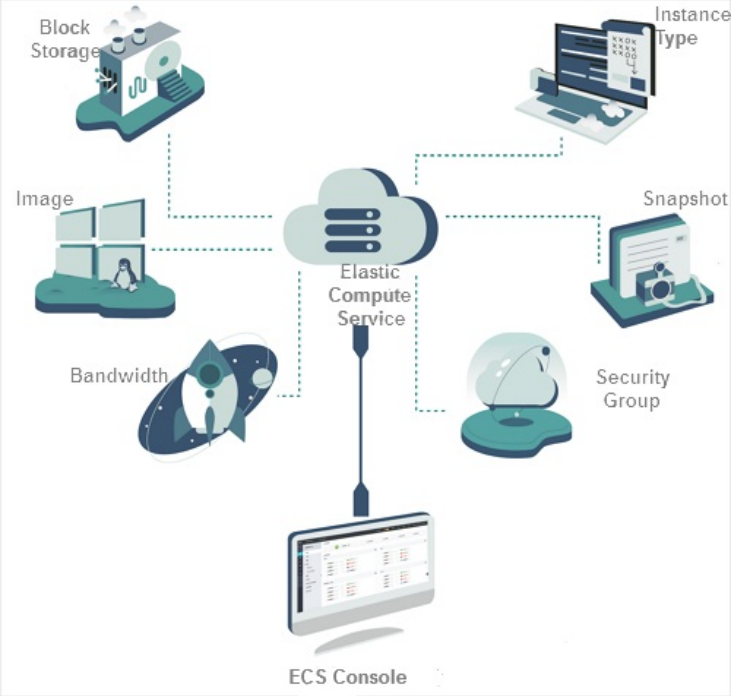
2.1.1. What is ECS?

2.1.1.1. Overview

Elastic Compute Service (ECS) is a type of computing service that features elastic processingcapabilities. Compared with physical servers, ECS can be more efficientlymanaged and is more user-friendly. You can create instances, resize disks, and add or release any number of ECS instances at any time based on your business needs.

An ECS instance is a virtual computing environment that contains the most basic components of computers such as the CPU, memory, and storage. Users perform operations on ECS instances. Instances are core components of ECS, and operations can be performed on instances by using the ECS console. Other resources such as block storage, images, and snapshots can only be used after they are integrated into ECS instances. For more information, see ECS components.

ECS components



2.1.1.2. Instance lifecycle

The lifecycle of an ECS instance begins when the instance is created and ends when the instance is released. This topic describes the instance states in the ECS console, state attributes, and corresponding instance states in API responses.

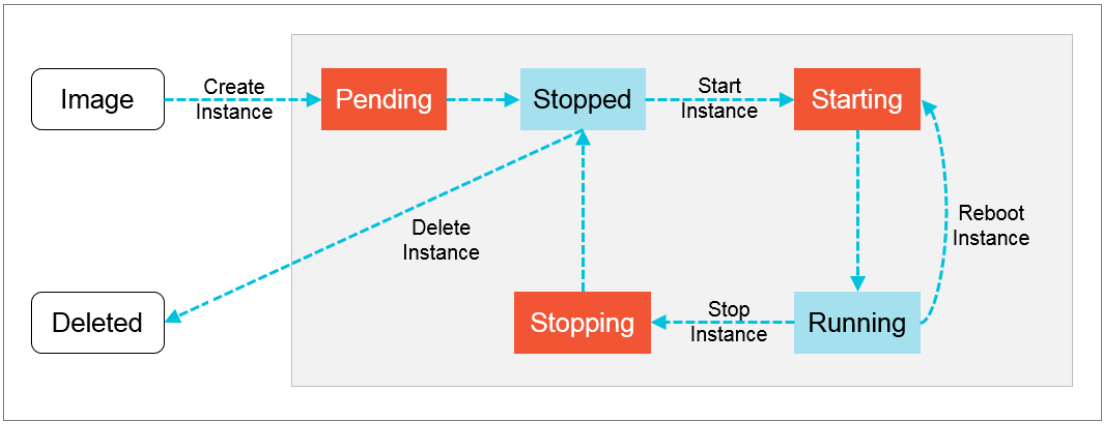
The following table describes the states that an ECS instances may go through during its lifecycle.

Instance states

State	State attribute	Description	State in an API response
Instance being created	Intermediate	The instance is being created and waiting to be started. If an instance remains in the Instance being created state for an extended period of time, an exception has occurred.	Pending
Starting	Intermediate	When you start or restart an instance by using the ECS console or calling an API operation, the instance enters this state before it enters the Running state. If an instance remains in the Starting state for an extended period of time, an exception has occurred.	Starting
Running	Stable	While an instance is in the Running state, the instance can function normally and can accommodate your business needs.	Running
Stopping	Intermediate	When you stop an instance by using the ECS console or calling an API operation, the instance enters this state before it enters the Stopped state. If an instance remains in the Stopping state for an extended period of time, an exception has occurred.	Stopping
Stopped	Stable	An instance enters this state when it is stopped. An instance in the Stopped state cannot provide external services.	Stopped
Reinitializing	Intermediate	When you re-initialize the system disk or a data disk of an instance by using the ECS console or calling an API operation, the instance enters this state before it enters the Running state. If an instance remains in the Reinitializing state for an extended period of time, an exception has occurred.	Stopped
Changing system disk	Intermediate	When you replace the system disk of an instance by using the ECS console or calling an API operation, the instance enters the Changing system disk state before it enters the Running state. If an instance remains in the Changing system disk state for an extended period of time, an exception has occurred.	Stopped

Instance states describes the relationships between instance states in the ECS console and instance states in API responses. The following figure shows the transitions between instance states in API responses.

Transitions between instance states in API responses



2.1.2. Instructions

2.1.2.1. Restrictions

Learn about restrictions before performing operations on ECS instances.

- Do not upgrade the kernel or operating system version of an ECS instance.
- Do not start SELinux for Linux systems except CentOS and RedHat.
- Do not detach PVDriver.
- Do not arbitrarily modify the MAC address of the network interface.

2.1.2.2. Suggestions

Consider the following suggestions to make more efficient use of ECS:

- ECS instances with 4 GiB or higher memory must use a 64-bit operating system. 32-bit operating systems have a maximum of 4 GiB of memory addressing.
- A 32-bit Windows operating system supports a maximum of 4 CPU cores.
- To ensure service continuity and avoid failover-induced service unavailability, we recommend that you configure service applications to boot automatically at system startup.

2.1.2.3. Limits

Before using ECS instances, you must be familiar with the limits of instance families.

General limits

- Windows operating systems support a maximum of 64 vCPUs in instance specifications.
- ECS instances do not support the installation of virtualization software and secondary virtualization.
- Sound card applications are not supported. Only GPU instances support virtual sound cards. External hardware devices, such as hardware dongles, USB flash drives, external hard disks, and bank U keys, cannot be directly connected to ECS instances.
- ECS does not support multicast protocols. If multicasting services are required, we recommend that you use unicast instead.

Instance family ga1

To create a ga1 instance, you must use one of the following images pre-installed with drivers:

- Ubuntu 16.04 with an AMD GPU driver pre-installed
- Windows Server 2016 English version with an AMD GPU driver pre-installed

- Windows Server 2008 R2 English version with an AMD GPU driver pre-installed

Note:

- A ga1 instance uses an optimized driver provided by Alibaba Cloud and AMD. The driver is installed in images provided by Alibaba Cloud and is currently unavailable for download.
- If the GPU driver malfunctions due to improper removal of related components, you need to replace the system disk to restore GPU related functions.

  > **Note** This operation causes data loss.

- If the driver malfunctions because an improper image is selected, you need to replace the system disk to reselect an image with an AMD GPU driver pre-installed.
- For Windows Server 2008 or earlier, you cannot connect to the VNC after the GPU driver takes effect. The VNC is irresponsive with a black screen or stuck at the splash screen. You can use other methods such as Remote Desktop Protocol (RDP) to access the system.
- RDP does not support DirectX, OpenGL, or other related applications. You need to install the VNC and a client, or use other supported protocols such as PCOIP and XenDesktop HDX 3D.

Instance families gn4, gn5i, and gn5

- Bandwidth: If you use an image of Windows Server 2008 R2 for a gn4 instance, you cannot use the Connect to VNC function in the ECS console to connect to the instance after the installed GPU driver takes effect. You need to set the bandwidth to a non-zero value or attach an Elastic IP address to the created instance.
- Image: If an NVIDIA GPU driver is not required, you can select any image, and then Install the CUDA and GPU drivers for a Linux instance or Install the CUDA and GPU drivers for a Windows instance.

2.1.2.4. Notice for Windows users

Before using Windows-based ECS instances, you must consider the following points:

- Data loss may occur if a local disk is used as the data disk of an instance. We recommend that you use a cloud disk to create your instance if you are not sure about the reliability of the data architecture.
- Do not close the built-in shutdownmon.exe process. Otherwise, the server may require a longer time to restart.
- Do not rename, delete, or disable Administrator accounts or it may affect the use of the server.
- We do not recommend that you use virtual memory.
- When you modify your computer name, you must synchronize the following key values in the registry. Otherwise, the computer name cannot be modified, causing failure when installing certain third-party programs. The following key values must be modified in the registry:

```
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\ComputerName\ActiveComputerName
```

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ComputerName\ComputerName
```

2.1.2.5. Notice for Linux users

Before using Linux-based ECS instances, you must consider the following points:

- Do not modify content of the default /etc/issue files under a Linux instance. Otherwise, the custom image created from the instance cannot be recognized, and instances created based on the image cannot start as expected.
- Do not arbitrarily modify the permissions of each directory in the partition where the root directory is located, especially permissions of /etc, /sbin, /bin, /boot, /dev, /usr, and /lib directories. Improper modification of permissions can cause errors.
- Do not rename, delete, or disable Linux root accounts.

- Do not compile or perform any other operations on the Linux kernel.
- We do not recommend the use of Swap for partitioning.
- Do not enable the NetWorkManager service. This service conflicts with the internal network service of the system, causing network errors.

2.1.2.6. Notice on defense against DDoS attacks

You need to purchase Anti-DDoS Pro to defend against DDoS attacks. For more information, see *Apsara Stack Security Product Introduction*.

2.1.3. Quick start

2.1.3.1. Overview

This topic describes how to quickly create and connect to an ECS instance.

Perform the following procedure:

1. Create a security group

   A security group is a virtual firewall used to control traffic to and from ECS instances. Each ECS instance must be added to at least one security group. Before creating an instance, you must select a security group to control traffic to and from the instance.

2. Create an instance

   An ECS instance is a virtual machine that contains basic computing components such as CPU, memory, operating system, network, and disks. After a security group is created, you can select an instance type based on your business requirements. For more information, see Instance types.

3. Connect to an instance

   Select a remote connection method based on the network configuration and operating system of the ECS instance and your local operating system. After you log on to the instance, you can perform other operations on it, such as installing applications.

2.1.3.2. Log on to the ECS console

This topic describes how to log on to the ECS console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

> **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
>
> - Uppercase or lowercase letters
> - Digits
> - Special characters, which include ! @ # $ %

3. Click **Login**.
4. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.

2.1.3.3. Create a security group

Security groups are an important means for network security isolation. They implement network access control for one or more ECS instances.

Prerequisites

A virtual compute cloud (VPC) is created. For more information, see *VPC User Guide*.

Context

Security groups determine whether the instances in the same account that are deployed within the same VPC and region can communicate with each other. By default, if the instances belong to the same security group, they can communicate with each other over the internal network. If the instances belong to different security groups, you can authorize mutual access between the security groups to allow the instances to communicate with each other over the internal network.

Procedure

1.
2.
3.
4. Click **Create Security Group**.
5. Configure the parameters listed in the following table to create a security group.

Type	Parameter	Required	Description
Region	Organization	Yes	Select an organization in which to create the security group. Make sure that the security group and the VPC belong to the same organization.
	Resource Set	Yes	Select a resource set in which to create the security group. Make sure that the security group and the VPC belong to the same resource set.
	Region	Yes	Select a region in which to create the security group. Make sure that the security group and VPC belong to the same region.
	Zone	Yes	Select a zone in which to create the security group.
	VPC	Yes	Select a VPC in which to create the security group.

Type	Parameter	Required	Description
Basic Settings	Security Group Name	No	Enter a name for the security group. The name must be 2 to 128 characters in length and start with a letter. It can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,). It cannot start with http:// or https://.
	Description	No	Enter a description for the security group. To simplify future management operations, we recommend that you provide an informational description. The description must be 2 to 256 characters in length and start with a letter. It can contain letters, digits, periods (.), underscores (_), hyphens (-), and commas (,). It cannot start with http:// or https://.

6. Click **Submit**.

2.1.3.4. Create an instance

An ECS instance is a virtual machine that contains the basic computing components of a server, such as CPU, memory, operating system, network, and disks.

Prerequisites

- A VPC and a vSwitch are created. For more information, see the "Create a VPC" and "Create a vSwitch" topics in *VPC User Guide*.
- If you want to assign an IPv6 address to the instance that you want to create, make sure that the VPC and vSwitch are associated with IPv6 CIDR blocks. For more information, see the "Enable an IPv6 CIDR block for a VPC" and "Enable an IPv6 CIDR block for a vSwitch" topics in *VPC User Guide*
- A security group is available. If no security group is available, create one. For more information, see Create a security group.

Context

Some limits apply when you create GPU-accelerated instances. For more information, see Limits.

Procedure

1.
2.
3.
4. Click **Create Instance**.
5. Configure the parameters listed in the following tables to create an instance.
   i. Configure the basic settings of the instance.

Parameter	Required	Description
Organization	Yes	Select an organization in which to create the instance.
Resource Set	Yes	Select a resource set in which to create the instance.

ii. Select a region and zone for the instance.

Parameter	Required	Description
Region	Yes	Select a region in which to create the instance.
Zone	Yes	Select a zone in which to create the instance. Zones are the physical zones with separate power supplies and networks within the same region. The internal networks of zones are interconnected, and faults in one zone are isolated from the other zones. To increase the availability of your applications, we recommend that you create instances in different zones.

iii. Configure the network of the instance.

Parameter	Required	Description
Network Type	Yes	Select the type of the network in which to create the instance. Valid value: **VPC**.
VPC	Yes	Select a VPC in which to create the instance.
vSwitch	Yes	Select a vSwitch to which to connect the instance.
Private IP Address	No	Specify a private IPv4 address for the instance. The private IPv4 address must be within the CIDR block of the vSwitch. If you do not specify a private IP address, the system automatically allocates a private IP address to the instance.

iv. (Optional)Specify whether to assign an IPv6 address to the instance.

v. Select a security group in which to create the instance.

vi. Select an instance family and instance type for the instance.

Parameter	Required	Description
Instance Family	Yes	Select an instance family for the instance. After you select an instance family, you must select an instance type.
Instance Type	Yes	Select an instance type for the instance. Instance types that have specific CPU and memory combinations do not support Windows Server images. For more information, see *ECS Product Introduction*.

vii. Configure the image to be used by the instance.

Parameter	Required	Description
Image Type	Yes	Select an image type. Valid values: **Public Image** and **Custom Image**.
Public Image	Subject to the image type	Select a public image for the instance. Public images provided by Alibaba Cloud are licensed, secure, and stable. Public images include Windows Server images and major Linux images. This parameter is required when you set Image Type to **Public Image**. When you use an image that supports DHCPv6 to create an instance, an IPv6 address is automatically assigned to the instance. The created instance can use this IPv6 address to communicate over the internal network. When you use an image that does not support DHCPv6 to create an instance, you must manually assign an IPv6 address to the instance. The following images support DHCPv6: ■ Linux images: ■ CentOS 7.6 IPv6 64-bit ■ CentOS 6.10 64-bit ■ SUSE Linux Enterprise Server 12 SP4 64-bit ■ Windows Server images **Note** To use an IPv6 address to communicate over the Internet, you must enable public bandwidth for the IPv6 address. For more information, see the "Enable Internet connectivity for an IPv6 address" topic in *VPC User Guide*.
Custom Image	Subject to the image type	Select a custom image for the instance. Custom images are created from instances or snapshots, or imported from your local device. This parameter is required when you set Image Type to **Custom Image**.

viii. Configure the storage settings for the instance.

Parameter	Required	Description
System Disk	Yes	Specify the disk category and capacity of the system disk. Valid values for the disk category: **Ultra Disk** and **Standard SSD**. The system disk capacity must range from 20 GiB to 500 GiB.
Data Disk	No	You can click Data Disk to add data disks. Specify the disk category and capacity of each data disk. Valid values for the disk category: **Ultra Disk** and **Standard SSD**. A maximum of 16 data disks can be added to an instance. The maximum capacity of each data disk is 32 TiB. You can select or clear **Release with Instance** and **Encryption** for each data disk. To encrypt a data disk, you can select **AES256** or **SM4** from the **Encryption Method** drop-down list and then select a key created in **Key Management Service (KMS)** from the **Encryption Key** drop-down list. You can also add data disks after the instance is created. For more information, see Create a disk.

ix. Configure the logon password settings for the instance.

Parameter	Required	Description
Password Setting	Yes	Specify when to set the password. Valid values: **Now** and **Later**. If Later is selected, you can use the password reset feature to set a password after the instance is created. For more information, see Change the logon password of an instance. **Note** The password is used to log on to the instance, not to a VNC management terminal.
Logon Password	No	Set the password to be used to log on to the instance. The password must be 8 to 30 characters in length and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The supported special characters include `( ) ` ~ ! @ # $ % ^ & * - _ + = | { } [ ] : ; ' < > , . ? /`
Confirm Password	No	Re-enter the password.

x. (Optional)Select a deployment set in which to create the instance.

xi. (Optional)Enter an instance name.

The name must be 2 to 128 characters in length and start with a letter. It can contain periods (.), underscores (_), colons (:), and hyphens (-).

If you do not specify a name, the system assigns an instance name at random.

xii. (Optional)In the Custom Script field, enter the user data to be automatically run on instance startup.

Windows supports batch and PowerShell scripts. Before you perform Base64 encoding of user data, make sure that the data to be encoded includes `[bat]` or `[powershell]` as the first line. Linux supports shell scripts.

xiii. Enter the number of instances that you want to create.

The number must be an integer ranging from 1 to 100.

6. Click **Submit**.

Result

The instance appears in the instance list. When the instance is being created, it is in the **Preparing** state. After the instance is created, it enters the **Running** state.

2.1.3.5. Connect to an instance

2.1.3.5.1. Instance connecting overview

After an instance is created, you can connect to the instance to perform operations such as installing applications.

You can use one of the following methods to connect to an instance:

- Use remote connection tools to connect to instances that have public IP addresses. For more information about the procedure, see the following topics:
  - Connect to a Linux-based instance by using SSH commands in Linux or Mac OS X
  - Connect to a Linux-based instance by using remote connection tools in Windows
  - Connect to a Windows-based instance by using RDP
- Use the VNC feature in the ECS console. For more information, see Connect to an instance by using a VNC management terminal.

The username of a Windows instance is Administrator, and that of a Linux instance is root.

2.1.3.5.2. Connect to a Linux instance by using SSH commands in Linux or Mac OS X

This topic describes how to use SSH commands to connect to a Linux instance.

Prerequisites

- The instance and the security group are created.
- The instance is in the **Running** state.
- A logon password is set for the instance.
- An Elastic IP address (EIP) is bound with the instance.
- An inbound security group rule is added to the security group to allow the SSH port.

Rule direction	Authorization policy	Protocol type	Port range	Priority	Authorization type	Authorization object
Inbound	Accept	TCP	22/22	1	IPv4 CIDR block	0.0.0.0/0

Procedure

1. Enter the following command and press the Enter key.

```
ssh root@instance IP
```

2. Enter the instance password of the root user and press the Enter key.

2.1.3.5.3. Connect to a Linux-based instance by using remote connection tools in Windows

This topic describes how to connect to an instance by using the PuTTY tool.

Prerequisites

Remote connection tools are designed with similar logics. In this example, PuTTY is used to connect to an instance. Download PuTTY at the following URL: .

Procedure

1. Download and install PuTTY for Windows.
2. Start the PuTTY client and complete the following settings:
   - Host Name (or IP Address): Enter the EIP of the instance to be connected.
   - Port: Select the default port 22.
   - Connection Type: Select SSH.
   - Saved Session: Enter the name of the session. Click **Save**. After the settings are saved, PuTTY remembers the name and IP address of the instance. This eliminates the need to enter them every time you connect to the instance.
3. Click **Open** to connect to the instance.

   When you connect to the instance for the first time, PuTTY displays security alerts. Click **Yes** to proceed.
4. Enter the username root and press Enter.
5. Enter the password for the instance and press Enter.

   If a message similar to the following one appears, a connection to the instance is established.

```
Welcome to aliyun Elastic Compute Server!
```

2.1.3.5.4. Connect to a Windows instance by using RDP

This topic describes how to connect to a Windows instance by using Remote Desktop Protocol (RDP).

Prerequisites

- The instance and the security group are created.
- The instance is in the **Running** state.
- A logon password is set for the instance.
- An Elastic IP address (EIP) is bound with the instance.

- An inbound security group rule is added to the security group to allow the RDP port.

Rule direction	Authorization policy	Protocol type	Port range	Priority	Authorization type	Authorization object
Inbound	Accept	TCP	3389/3389	1	IPv4 CIDR block	0.0.0.0/0

- CredSSP-related security updates are installed on the operating system of the instance.

Procedure

1. Activate the Remote Desktop Connection feature by using any of the following methods:
   - Click **Start**, enter *mstsc* in the search box, and click **mstsc** in the search result.
   - Press Windows Key + R. In the **Run** dialog box that appears, enter *mstsc* and click **OK**.
2. In the **Remote Desktop Connection** dialog box, enter the EIP of the instance and click **Show Options**.
3. Enter the username.

   The default username is administrator.
4. (Optional)If you do not want to enter the password upon subsequent logons, select **Allow me to save credentials**.
5. Click **Connect**.
6. In the **Windows Security** dialog box that appears, enter the password for the account and click **OK**.

Result

After you log on to the instance, the Windows desktop appears.

If authentication errors occur or the required function is not supported, install security updates.

1. Connect to an ECS instance by using the VNC before proceeding.
2. Choose **Start > Control Panel**.
3. Click **System and Security**.
4. Click **Check for updates** in the **Windows Updates** pane.
5. If updates are available, click **Install updates**.
6. Restart the instance.

2.1.3.5.5. Connect to an instance by using a VNC management terminal

If other remote connection tools such as PuTTy, Xshell, and SecureCRT are not installed or do not work properly, you can access your instances by using a VNC management terminal in the ECS console.

Prerequisites

- The instance to which you want to connect is in the **Running** state.
- The root certificate is imported to your web browser. For more information, see Install the certificate for VNC in Windows.
- The VNC password is reset if it is your first time to connect to the instance after the instance is created. For more information, see Change the VNC password.

Context

The VNC password is used to log on to a VNC management terminal in the ECS console, whereas the instance password is used to log on to the instance.

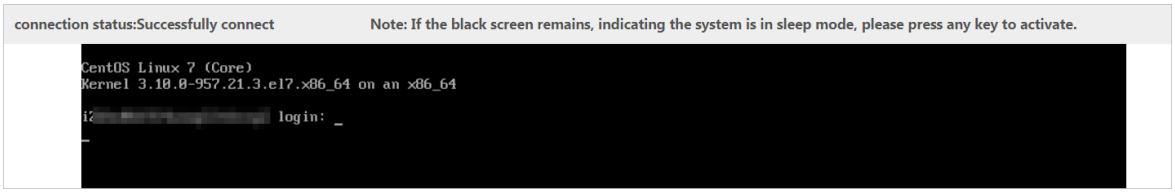
You can use a VNC management terminal to connect to an instance to solve specific issues. The following table lists some of the issues.

Issue	Solution
The instance starts slowly due to self-check on startup.	Check the self-check progress.
The firewall of the operating system is enabled by mistake.	Disable the firewall.
Abnormal processes appear and consume large amounts of CPU or bandwidth resources.	Troubleshoot and terminate the abnormal processes.

Procedure

1.
2.
3.
4. Find the instance to which you want to connect and click **Remote Connection** in the **Actions** column.
5. Enter the VNC password and click **OK**.

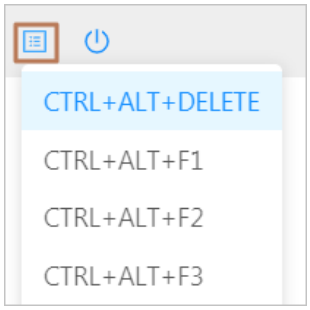
   After you are logged on to the VNC management terminal, a logon page similar to the following one appears.



6. Enter your username and password.
   - Fo a Linux instance, enter the username *root* and the logon password.

     > **Note** Passwords in Linux are not displayed as you type. Press the Enter key after you enter the password.

   - For a Windows instance, to use a key combination such as Ctrl+Alt+Delete, click the List icon in the upper-right corner of the VNC page and select the corresponding key combination from the drop-down list.



   Enter the username and password as prompted, and click the Log On icon such as  .

2.1.4. Instances

2.1.4.1. Create an instance

An ECS instance is a virtual machine that contains the basic computing components of a server, such as CPU, memory, operating system, network, and disks.

Prerequisites

- A VPC and a vSwitch are created. For more information, see the "Create a VPC" and "Create a vSwitch" topics in *VPC User Guide*.
- If you want to assign an IPv6 address to the instance that you want to create, make sure that the VPC and vSwitch are associated with IPv6 CIDR blocks. For more information, see the "Enable an IPv6 CIDR block for a VPC" and "Enable an IPv6 CIDR block for a vSwitch" topics in *VPC User Guide*
- A security group is available. If no security group is available, create one. For more information, see Create a security group.

Context

Some limits apply when you create GPU-accelerated instances. For more information, see Limits.

Procedure

1.
2.
3.
4. Click **Create Instance**.
5. Configure the parameters listed in the following tables to create an instance.
   i. Configure the basic settings of the instance.

Parameter	Required	Description
Organization	Yes	Select an organization in which to create the instance.
Resource Set	Yes	Select a resource set in which to create the instance.

   ii. Select a region and zone for the instance.

Parameter	Required	Description
Region	Yes	Select a region in which to create the instance.
Zone	Yes	Select a zone in which to create the instance. Zones are the physical zones with separate power supplies and networks within the same region. The internal networks of zones are interconnected, and faults in one zone are isolated from the other zones. To increase the availability of your applications, we recommend that you create instances in different zones.

iii. Configure the network of the instance.

Parameter	Required	Description
Network Type	Yes	Select the type of the network in which to create the instance. Valid value: **VPC**.
VPC	Yes	Select a VPC in which to create the instance.
vSwitch	Yes	Select a vSwitch to which to connect the instance.
Private IP Address	No	Specify a private IPv4 address for the instance. The private IPv4 address must be within the CIDR block of the vSwitch. If you do not specify a private IP address, the system automatically allocates a private IP address to the instance.

iv. (Optional)Specify whether to assign an IPv6 address to the instance.

v. Select a security group in which to create the instance.

vi. Select an instance family and instance type for the instance.

Parameter	Required	Description
Instance Family	Yes	Select an instance family for the instance. After you select an instance family, you must select an instance type.
Instance Type	Yes	Select an instance type for the instance. Instance types that have specific CPU and memory combinations do not support Windows Server images. For more information, see *ECS Product Introduction*.

vii. Configure the image to be used by the instance.

Parameter	Required	Description
Image Type	Yes	Select an image type. Valid values: **Public Image** and **Custom Image**.
Public Image	Subject to the image type	Select a public image for the instance. Public images provided by Alibaba Cloud are licensed, secure, and stable. Public images include Windows Server images and major Linux images. This parameter is required when you set Image Type to **Public Image**. When you use an image that supports DHCPv6 to create an instance, an IPv6 address is automatically assigned to the instance. The created instance can use this IPv6 address to communicate over the internal network. When you use an image that does not support DHCPv6 to create an instance, you must manually assign an IPv6 address to the instance. The following images support DHCPv6: ■ Linux images: ■ CentOS 7.6 IPv6 64-bit ■ CentOS 6.10 64-bit ■ SUSE Linux Enterprise Server 12 SP4 64-bit ■ Windows Server images **Note** To use an IPv6 address to communicate over the Internet, you must enable public bandwidth for the IPv6 address. For more information, see the "Enable Internet connectivity for an IPv6 address" topic in *VPC User Guide*.
Custom Image	Subject to the image type	Select a custom image for the instance. Custom images are created from instances or snapshots, or imported from your local device. This parameter is required when you set Image Type to **Custom Image**.

viii. Configure the storage settings for the instance.

Parameter	Required	Description
System Disk	Yes	Specify the disk category and capacity of the system disk. Valid values for the disk category: **Ultra Disk** and **Standard SSD**. The system disk capacity must range from 20 GiB to 500 GiB.
Data Disk	No	You can click Data Disk to add data disks. Specify the disk category and capacity of each data disk. Valid values for the disk category: **Ultra Disk** and **Standard SSD**. A maximum of 16 data disks can be added to an instance. The maximum capacity of each data disk is 32 TiB. You can select or clear **Release with Instance** and **Encryption** for each data disk. To encrypt a data disk, you can select **AES256** or **SM4** from the **Encryption Method** drop-down list and then select a key created in **Key Management Service (KMS)** from the **Encryption Key** drop-down list. You can also add data disks after the instance is created. For more information, see Create a disk.

ix. Configure the logon password settings for the instance.

Parameter	Required	Description
Password Setting	Yes	Specify when to set the password. Valid values: **Now** and **Later**. If Later is selected, you can use the password reset feature to set a password after the instance is created. For more information, see Change the logon password of an instance. **Note** The password is used to log on to the instance, not to a VNC management terminal.
Logon Password	No	Set the password to be used to log on to the instance. The password must be 8 to 30 characters in length and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The supported special characters include `( ) ` ~ ! @ # $ % ^ & * - _ + = | { } [ ] : ; ' < > , . ? /`
Confirm Password	No	Re-enter the password.

x. (Optional)Select a deployment set in which to create the instance.

xi. (Optional)Enter an instance name.

The name must be 2 to 128 characters in length and start with a letter. It can contain periods (.), underscores (_), colons (:), and hyphens (-).

If you do not specify a name, the system assigns an instance name at random.

xii. (Optional)In the Custom Script field, enter the user data to be automatically run on instance startup.

Windows supports batch and PowerShell scripts. Before you perform Base64 encoding of user data, make sure that the data to be encoded includes `[bat]` or `[powershell]` as the first line. Linux supports shell scripts.

xiii. Enter the number of instances that you want to create.

The number must be an integer ranging from 1 to 100.

6. Click **Submit**.

Result

The instance appears in the instance list. When the instance is being created, it is in the **Preparing** state. After the instance is created, it enters the **Running** state.

2.1.4.2. Connect to an instance

2.1.4.2.1. Instance connecting overview

After an instance is created, you can connect to the instance to perform operations such as installing applications.

You can use one of the following methods to connect to an instance:

- Use remote connection tools to connect to instances that have public IP addresses. For more information about the procedure, see the following topics:
  - Connect to a Linux-based instance by using SSH commands in Linux or Mac OS X
  - Connect to a Linux-based instance by using remote connection tools in Windows
  - Connect to a Windows-based instance by using RDP
- Use the VNC feature in the ECS console. For more information, see Connect to an instance by using a VNC management terminal.

The username of a Windows instance is Administrator, and that of a Linux instance is root.

2.1.4.2.2. Connect to a Linux-based instance by using SSH commands in Linux or Mac OS X

This topic describes how to use SSH commands to connect to a Linux-based instance.

Prerequisites

Create a security group and an instance.

Procedure

1. Enter the following command: `ssh root@instance IP`.
2. Enter the password for the *root* user to log on to the instance.

2.1.4.2.3. Connect to a Linux-based instance by using remote connection tools in Windows

This topic describes how to connect to an instance by using the PuTTY tool.

Prerequisites

Remote connection tools are designed with similar logics. In this example, PuTTY is used to connect to an instance. Download PuTTY at the following URL: .

Procedure

1. Download and install PuTTY for Windows.
2. Start the PuTTY client and complete the following settings:
   - Host Name (or IP Address): Enter the EIP of the instance to be connected.
   - Port: Select the default port 22.
   - Connection Type: Select SSH.
   - Saved Session: Enter the name of the session. Click **Save**. After the settings are saved, PuTTY remembers the name and IP address of the instance. This eliminates the need to enter them every time you connect to the instance.
3. Click **Open** to connect to the instance.

   When you connect to the instance for the first time, PuTTY displays security alerts. Click **Yes** to proceed.
4. Enter the username root and press Enter.
5. Enter the password for the instance and press Enter.

   If a message similar to the following one appears, a connection to the instance is established.

   ```
   Welcome to aliyun Elastic Compute Server!
   ```

2.1.4.2.4. Connect to a Windows instance by using RDC

This topic describes how to connect to a Windows instance by using Remote Desktop Connection (RDC).

Prerequisites

- A security group and a Windows instance are created.
- The instance is in the **Running** state.
- A logon password is set for the instance.
- An Elastic IP address is associated with the instance.
- An inbound security group rule is added to the security group to allow traffic on the RDP port.

Rule direction	Action	Protocol	Port range	Priority	Authorization type	Authorization object
Inbound	Allow	tcp	3389/3389	1	IPv4 addresses	0.0.0.0/0

Procedure

1. Use one of the following methods to enable RDC:
   - Click **Start**, enter *mstsc* in the search box, and click **mstsc** in the search result.

- Press the Windows logo key+R. In the **Run** dialog box that appears, enter *mstsc* and click **OK**.

2. In the **Remote Desktop Connection** dialog box, enter the Elastic IP address of the instance and click **Show Options**.
3. Enter the username.

   The default username is administrator.
4. (Optional)If you do not want to enter the password upon subsequent logons, select **Allow me to save credentials**.
5. Click **Connect**.
6. In the **Windows Security** dialog box that appears, enter the password corresponding to the username you entered and click **OK**.

Result

If the Windows desktop appears, a connection to the Windows instance is established.

If an error message is returned indicating that an authentication error has occurred and the function requested is not supported, install CredSSP updates and try again. Follow these steps to install the updates:

1. Connect to an ECS instance by using the VNC.
2. Choose **Start > Control Panel**.
3. Click **System and Security**.
4. Click **Check for updates** in the **Windows Updates** section.
5. If updates are available, click **Install updates**.
6. Restart the instance.

2.1.4.2.5. Install the certificate for VNC in Windows

Before you log on to the VNC management terminal, you must export the relative certificate from the site such as the Apsara Uni-manager Management Console and install the certificate in your local browser.

Context

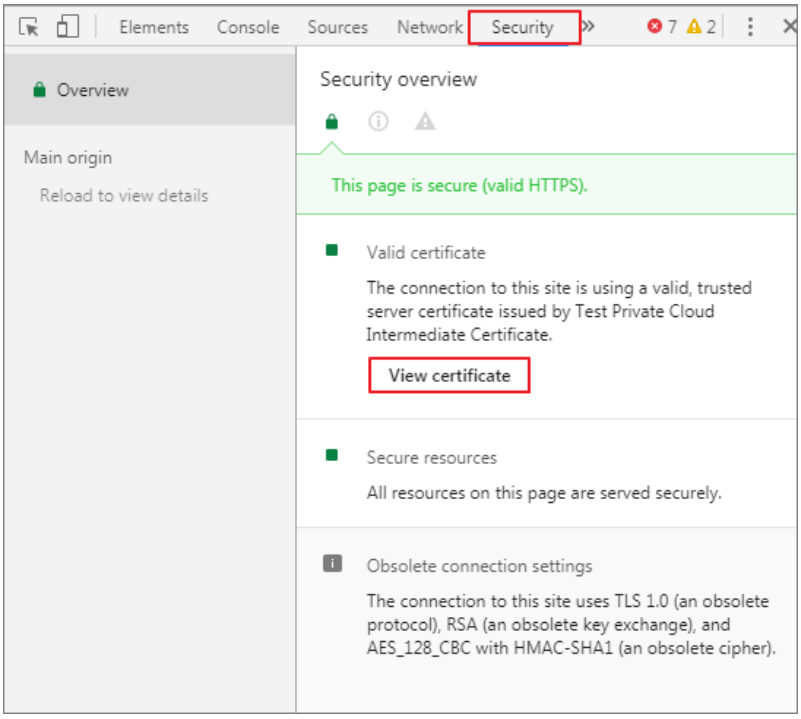
The VNC feature is provided by the VNC proxy service. The VNC proxy service uses a certificate different from that of Apsara Infrastructure Management Framework. The certificate of the VNC proxy service must be manually imported.

Procedure

1. Export the certificate from the Apsara Uni-manager Management Console.

i. Log on to the Apsara Uni-manager Management Console. Press the **F12** key or **Fn+F12** to view and select the certificate.

For example, in the Chrome browser, press the **F12** key to open Chrome DevTools.



ii. In the **Certificate** dialog box, click the **Certificate Path** tab, select the root certificate, and then click **View Certificate**.

iii. In the **Certificate** dialog box, click the **Details** tab and then click **Copy to File**.

iv. In the **Certificate Export Wizard** dialog box, click **Next**.

v. Select **DER encoded binary X.509 (.CER)** as the format and then click **Next**.

vi. Click **Browse**, choose where to store the certificate, enter a file name, and then click **Save**.

vii. Click **Next**.

viii. Click **Finish**.

ix. Click **OK**.

2. Install the certificate in your local browser.

i. Double-click the certificate.

ii. In the **Certificate** dialog box, click **Install Certificate**.

iii. In the **Certificate Import Wizard** dialog box, click **Next**.

iv. Select **Place all certificates in the following store** and click **Browse**.

v. In the **Select Certificate Store** dialog box, select **Trusted Root Certificate Authority** and then click **OK**.

vi. In the **Certificate Import Wizard** dialog box, click **Next**.

vii. Click **Finish**.

viii. If a security warning message is displayed, click **Yes**.

3. Restart your browser and log on to the Apsara Uni-manager Management Console.

After the certificate is installed, the security warning message is no longer displayed on the left of the URL when you log on to the Apsara Uni-manager Management Console.

2.1.4.2.6. Connect to an instance by using a VNC management terminal

If other remote connection tools such as PuTTy, Xshell, and SecureCRT are not installed or do not work properly, you can access your instances by using a VNC management terminal in the ECS console.

Prerequisites

- The instance to which you want to connect is in the **Running** state.
- The root certificate is imported to your web browser. For more information, see Install the certificate for VNC in Windows.
- The VNC password is reset if it is your first time to connect to the instance after the instance is created. For more information, see Change the VNC password.

Context

The VNC password is used to log on to a VNC management terminal in the ECS console, whereas the instance password is used to log on to the instance.

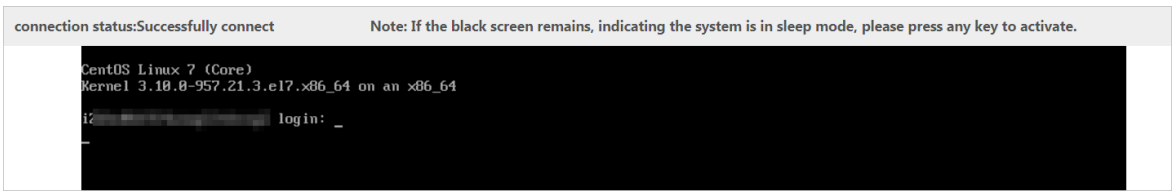
You can use a VNC management terminal to connect to an instance to solve specific issues. The following table lists some of the issues.

Issue	Solution
The instance starts slowly due to self-check on startup.	Check the self-check progress.
The firewall of the operating system is enabled by mistake.	Disable the firewall.
Abnormal processes appear and consume large amounts of CPU or bandwidth resources.	Troubleshoot and terminate the abnormal processes.

Procedure

1.
2.
3.
4. Find the instance to which you want to connect and click **Remote Connection** in the **Actions** column.
5. Enter the VNC password and click **OK**.

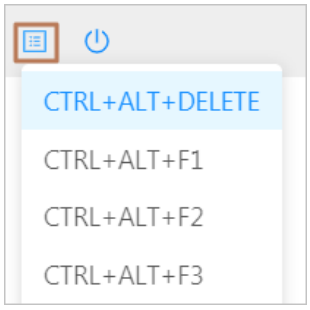
   After you are logged on to the VNC management terminal, a logon page similar to the following one appears.



6. Enter your username and password.
   - Fo a Linux instance, enter the username *root* and the logon password.

> **Note** Passwords in Linux are not displayed as you type. Press the Enter key after you enter the password.

- For a Windows instance, to use a key combination such as Ctrl+Alt+Delete, click the List icon in the upper-right corner of the VNC page and select the corresponding key combination from the drop-down list.



Enter the username and password as prompted, and click the Log On icon such as  .

2.1.4.3. View instances

You can view the list of instances and the details of individual instances. The details of an instance include basic configurations, disks, snapshots, security groups, and elastic network interfaces (ENIs).

Procedure

1.
2.
3.
4. Select a filter option from the drop-down list, enter the relevant information in the search bar, and click **Search**.

   You can select multiple filter options to narrow down search results.

Filter option	Description
Instance Name	Enter an instance name to search for the instance.
Instance ID	Enter an instance ID to search for the instance.
IP Address	Enter the IP address of an instance to search for the instance.
VPC ID	Enter a VPC ID to search for instances that belong to the VPC.
Image ID	Enter an image ID to search for instances that use the image.
Status	Select an instance status to search for instances in that status. Valid values: ○ **Running** ○ **Stopped** ○ **Starting** ○ **Stopped**
Security Group ID	Enter a security group ID to search for instances that belong to the security group.

Filter option	Description
Operating System	Enter the name of operating system to search for instances that use the operating system.

5. Use one of the following methods to go to the Instance Details page of an instance:
   - In the **Instance ID/Name** column, click the instance ID.
   - Click **Manage** in the **Actions** column corresponding to the instance.
   - Choose **More > Show Details** in the **Actions** column corresponding to the instance.

2.1.4.4. Modify an instance

You can modify the names, descriptions, and user data of created instances.

Procedure

1.
2.
3.
4. Find the instance that you want to modify and choose **More > Modify** in the **Actions** column.
5. Modify the name, description, and user data of the instance.

   The name must be 2 to 128 characters in length. The description must be 2 to 256 characters in length. The user data must be 2 to 999 characters in length.
6. Click **OK**.

2.1.4.5. Stop an instance

You can stop instances that are not in use. The stop operation interrupts the services that are running on the instances. Exercise caution when you perform this operation.

Prerequisites

The instance that you want to stop is in the **Running** state.

Procedure

1.
2.
3.
4. Use one of the following methods to stop instances:
   - To stop a single instance, find the instance and choose **More > Instance Status > Stop** in the **Actions** column.
   - To stop one or more instances at a time, select the instances and click **Stop** in the lower-left corner of the Instances page.
5. Click **OK**.

Result

When the instance is being stopped, its status in the **Status** column changes from **Running** to **Stopping**. After the instance is stopped, its status changes to **Stopped**.

2.1.4.6. Start an instance

You can start stopped instances.

Prerequisites

The instance that you want to start is in the **Stopped** state.

Procedure

1.
2.
3.
4. Use one of the following methods to start instances:
   - To start a single instance, find the instance and choose **More > Instance Status > Start** in the **Actions** column.
   - To start one or more instances at a time, select the instances and click **Start** in the lower-left corner of the Instances page.
5. Click **OK**.

Result

When the instance is being started, its status in the **Status** column changes from **Stopped** to **Starting**. After the instance is started, its status changes to **Running**.

2.1.4.7. Restart an instance

You must restart instances after you change their logon passwords or install system updates for the instances. The restart operation stops the instances for a period of time. This causes the services that are running on the instances to be interrupted. Exercise caution when you perform this operation.

Prerequisites

The instance that you want to restart is in the **Running** state.

Procedure

1.
2.
3.
4. Use one of the following methods to restart instances:
   - To restart a single instance, find the instance and choose **More > Instance Status > Restart** in the **Actions** column.
   - To restart one or more instances at a time, select the instances and click **Restart** in the lower-left corner of the Instances page.
5. In the Restart Instance dialog box, select a restart mode.
   - **Restart**: restarts the instances normally.
   - **Force Restart**: forcibly restarts the instances. This may result in loss of unsaved data.
6. Click **OK**.

2.1.4.8. Delete an instance

You can delete instances that are no longer needed to release their resources. Deleted instances cannot be recovered. We recommend that you back up data before you delete instances. If data disks are released along with instances, the data on the disks cannot be recovered.

Prerequisites

The instance to be deleted is in the **Stopped** state.

Procedure

1.
2.
3.
4. Select the instance that you want to delete and click **Delete** in the lower-left corner of the Instances page.
5. Click **OK**.

2.1.4.9. Change the instance type of an instance

You can change the instance types of instances to suit your business needs. This eliminates the need to create new instances.

Prerequisites

The instance whose instance type you want to change is in the **Stopped** state.

Procedure

1.
2.
3.
4. Find the instance whose instance type you want to change and click **Upgrade/Downgrade** in the **Actions** column.
5. On the Change Configurations page, select a new instance type and click **Submit**.

   The Change Configurations page shows the instance types available for selection.
6. Restart the instance by using the console or by calling an API operation for the new instance type to take effect.

   For more information, see Start an instance or the "StartInstance" topic in *ECS Developer Guide*.

2.1.4.10. Change the logon password of an instance

If you did not set a logon password when you created an instance or have forgotten the password, you can reset the password.

Procedure

1.
2.
3.
4. Find the instance whose logon password you want to change and use one of the following methods to go to the Instance Details page.
   - In the **Instance ID/Name** column, click the ID of the instance.
   - Click **Manage** in the **Actions** column.
   - In the **Actions** column, choose **More > Show Details**.
5. Click **Change Password**.
6. Enter and confirm the new password, and then click **OK**.

The password must be 8 to 30 characters in length and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The supported special characters include `( ) ' ~ ! @ # $ % ^ & * - _ + = | { } [ ] : ; ' < > , . ? /`

7. Restart the instance by using the ECS console or by calling an API operation for the new password to take effect.

   For more information, see Restart an instance or the "RebootInstance" topic in *ECS Developer Guide*.

2.1.4.11. Change the VNC password

If you log on to the VNC management terminal for the first time or forget the VNC password, you can reset the password.

Procedure

1.
2.
3.
4. Find the instance whose VNC password you want to change and use one of the following methods to go to the Instance Details page:
   - In the **Instance ID/Name** column, click the instance ID.
   - Click **Manage** in the **Actions** column.
   - In the **Actions** column, choose **More > Show Details**.
5. Click **Change VNC Password**.
6. Enter and confirm the new password, and click **OK**.

   The password must be 6 characters in length and can contain digits and letters. It cannot contain special characters.
7. Restart the instance by using the ECS console or by calling an API operation for the new password to take effect.

   For more information, see Restart an instance or the "RebootInstance" topic in *ECS Developer Guide*.

2.1.4.12. Add an instance to a security group

You can add created instanced to security groups and use security group rules to control network access for the instances.

Context

Security groups act as virtual firewalls to provide security isolation and implement network access control for instances.

Security groups determine whether the instances in the same account that are deployed within the same VPC and region can communicate with each other. By default, if the instances belong to the same security group, they can communicate with each other over the internal network. If the instances belong to different security groups, you can authorize mutual access between the security groups to allow the instances to communicate with each other over the internal network.

Procedure

1.
2.
3.
4. Find the security group to which you want to add an instance and click **Manage Instances** in the **Actions**

column.

5. Click **Add Instance**.
6. In the Add Instance dialog box, select an instance and click **OK**.

   An instance can belong to up to five security groups. After an instance is added to a security group, the rules of the security group automatically apply to the instance.

2.1.4.13. Customize instance data

ECS allows you to run the instance customization script upon startup and import data into instances.

Context

The instance data customization feature is applicable to both Windows and Linux instances. It allows you to:

- Run the instance customization script upon startup.
- Import data into instances.

Usage instructions

- **Limits**

  The instance data customization feature can only be used when an instance meets all the following requirements:

  - Network type: VPC
  - Image: a system image or a custom image that is inherited from the system image
  - Operating system: one type included in Supported operating systems Supported operating systems

Windows	Linux
■ Windows Server 2016 64-bit ■ Windows Server 2012 64-bit ■ Windows Server 2008 64-bit	■ CentOS ■ Ubuntu ■ SUSE Linux Enterprise ■ OpenSUSE ■ Debian ■ Aliyun Linux

  - When you configure instance data customization scripts, you must enter custom data based on the type of operating system and script.

    > **Note** Only English characters are allowed.

    - If your data is Base64 encoded, select **Enter Base64 Encoded Information**.

      > **Note** The size of the customization script cannot exceed 16 KB before the data is Base64 encoded.

    - For Linux instances, the script format must meet the requirements described in Types of Linux instance customization scripts.
    - For Windows instances, the script can only start with `[bat]` or `[powershell]`.

- After starting an instance, run a command to view the following information:
  - Execution result of the instance customization script
  - Data imported to instances

- **Console**: You can modify the custom instance data in the console. Whether the modified instance customization script needs to be re-executed depends on the script type. For example, if the `bootcmd` script in Cloud Config is modified for Linux instances, the script is automatically executed each time instances are restarted.
- **OpenAPI**: You can also use OpenAPI to customize instance data. For more information, see **CreateInstance** and **ModifyInstanceAttribute** in *ECS Developer Guide* .

Linux instance data customization scripts

Linux instance data customization scripts provided by Alibaba Cloud are designed based on the cloud-init architecture. They are used to automatically configure parameters of Linux instances. Customization script types are compatible with the cloud-init.

Description of Linux instance data customization scripts

- Linux instance customization scripts are executed after instances are started and before `/etc/init` is executed.
- Linux instance customization scripts can only be executed with root permissions by default.

Types of Linux instance customization scripts

- **User-Data Script**
  - Description: A script, such as shell script, is used to customize data.
  - Format: The first line must include `#!`, such as `#! /bin/sh` .
  - Limit: The script size (including the first line) cannot exceed 16 KB before the data is Base64 encoded.
  - Frequency: The script is executed only when instances are started for the first time.
  - Example:

```
#! /bin/sh
echo "Hello World. The time is now $(date -R)!" | tee /root/output10.txt
```

- **Cloud Config Data**
  - Description: Predefined data is used to configure services, such as specifying yum sources or importing SSH keys.
  - Format: The first line must be `#cloud-config` .
  - Limit: The script size (including the first line) cannot exceed 16 KB before the data is Base64 encoded.
  - Frequency: The script execution frequency varies with the specific service.
  - Example:

```
#cloud-config
apt:
primary:
- arches: [default]
uri: http://us.archive.ubuntu.com/ubuntu/
```

- **Include**
  - Description: The configuration content can be saved in a text file and imported into cloud-init as a URL.
  - Format: The first line must be `#include` .
  - Limit: The script size (including the first line) cannot exceed 16 KB before the data is Base64 encoded.
  - Frequency: The script execution frequency depends on the script type in the text file.
  - Example:

```
#include
http://ecs-image-test.oss-cn-hangzhou.aliyuncs.com/userdata/cloudconfig
```

- **GZIP format**
  - Description: Cloud-ini limits the size of customization scripts to 16 KB. You can compress and import the script file into the customization script if the file size exceeds 16 KB.
  - Format: The .gz file is imported into the customization script as a URL in `#include`.
  - Frequency: The script execution frequency depends on the script content contained in the GZIP file.
  - Example:

```
#include
http://ecs-image-test.oss-cn-hangzhou.aliyuncs.com/userdata/config.gz
```

View the custom data of a Linux instance

Run the following command in the instance:

```
curl http://100.100.100.200/latest/user-data
```

Windows instance customization scripts

Windows instance customization scripts independently developed by Alibaba Cloud can be used to initialize Windows instances.

There are two types of Windows instance customization scripts:

- Batch processing program: starts with `[bat]` and serves as the first line. The script size must be smaller than 16 KB before the data is Base64 encoded.
- PowerShell script: starts with `[powershell]` and serves as the first line. The script size must be smaller than 16 KB before the data is Base64 encoded.

View the custom data of a Windows instance

Run the following PowerShell command in the instance:

```
Invoke-RestMethod http://100.100.100.200/latest/user-data/
```

2.1.4.14. Change the private IP address of an instance

Each instance is assigned a private NIC and associated with a private IP address. You can change the private IP address of an instance. The new private IP address must be within the CIDR block of the vSwitch to which the instance is connected and must not be in use by another instance.

Prerequisites

The instance whose private IP address you want to change is in the **Stopped** state.

Procedure

1.
2.
3.
4. Find the instance whose private IP address you want to change and choose **More > Change Private IP Address** in the **Actions** column.
5. Enter a new private IP address.

   The new private IP address must be within the CIDR block of the vSwitch to which the instance is connected. This IP address must not be in use by another instance or be reserved for a specific purpose.

For example, if the CIDR block of the vSwitch is 192.168.1.0/24, you can use an IP address within the range from 192.168.1.1 to 192.168.1.254. The first address 192.168.1.0 is reserved to identify the subnet itself, and the last address 192.168.1.255 is reserved as the broadcast address. Neither address can be used.

6. Click **OK**.

2.1.4.15. Install the CUDA and GPU drivers for a Linux instance

You must install a GPU driver on GPU instances to use the GPU. If the image you use does not contain a pre-installed GPU driver, you must manually install the CUDA and GPU drivers for the instance.

Prerequisites

If your instance cannot connect to the Internet, the installation file cannot be downloaded. You can install an FTP client on the instance to transfer the installation file to the instance.

Context

When installing NVIDIA drivers, you must install the kernel package that contains the kernel header file before you install the CUDA and GPU drivers on the instance.

Procedure

1. Install the kernel package.
   i. Run the **uname -r** command to view the current kernel version.

      A similar output is displayed:

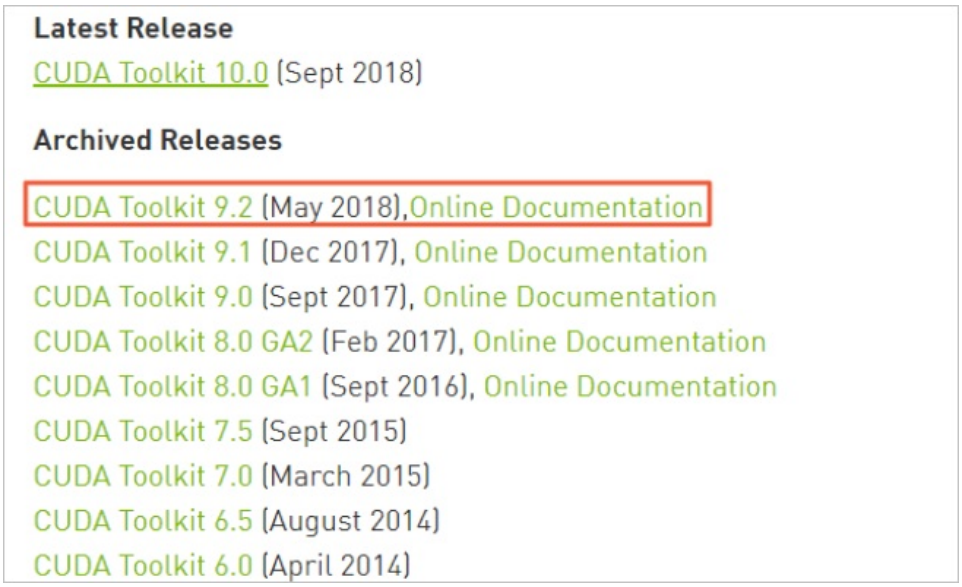
      - CentOS: `3.10.0-862.14.4.el7.x86_64`
      - Ubuntu: `4.4.0-117-generic`

   ii. Copy the kernel package of the corresponding version to the instance and install the package.
      - CentOS: Copy the RPM package of the `kernel-devel` component and run the **rpm -ivh 3.10.0-862.14.4.el7.x86_64.rpm** command to install the package. *3.10.0-862.14.4.el7.x86_64.rpm* is used as an example. Replace it with the actual package name.
      - Ubuntu: Copy the DEB package of the `linux-headers` component and run the **dpkg -i 4.4.0-117-generic.deb** command to install the package. *4.4.0-117-generic.deb* is used as an example. Replace it with the actual package name.
2. Download the CUDA Toolkit.

i. Access the official CUDA download page. Choose the version based on the GPU application requirements for CUDA.

This example uses CUDA Toolkit 9.2.

Download the CUDA Toolkit



ii. Choose a platform based on your operating system. Select **Installer Type** to **runfile (local)** and click **Download**.

NVIDIA drivers are already included in the CUDA Toolkit.

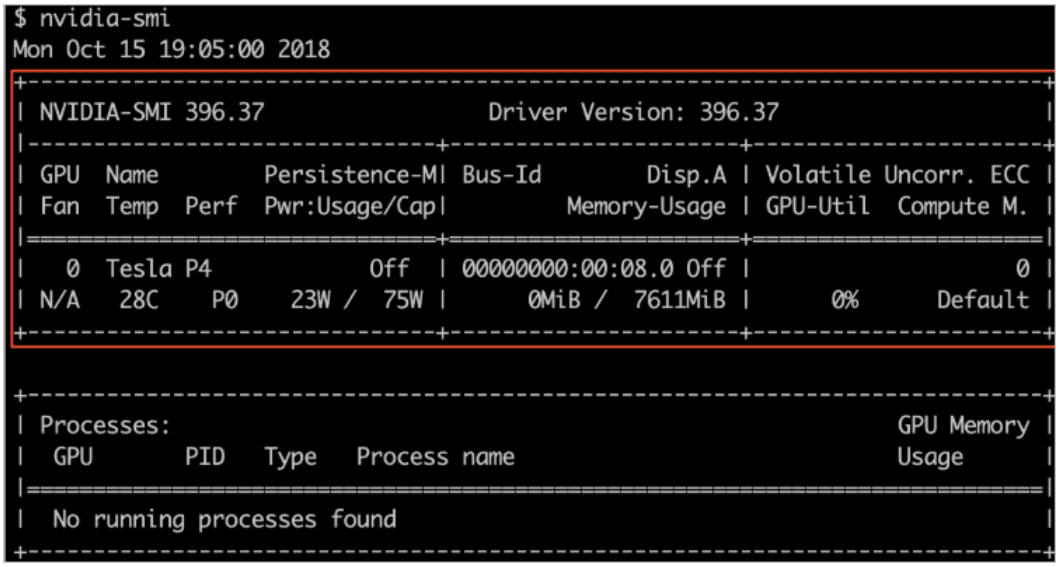
Download the drivers

3. Copy the downloaded *cuda_9.2.148_396.37_linux.run* file to the instance. *cuda_9.2.148_396.37_linux.run* is used as an example. Replace it with the actual file name.
4. Run the **sudo sh ./cuda_9.2.148_396.37_linux.run --silent --verbose --driver --toolkit --samples** command to install the CUDA driver. *cuda_9.2.148_396.37_linux.run* is used as an example. Replace it with the actual file name.
   The installation takes about 10 to 20 minutes. When `Driver: Installed` is displayed, the installation is successful.

   View the CUDA installation result

5. Run the **nvidia-smi** command to view the GPU driver status.
   If the output displays the details of the GPU driver, the driver is running properly.

   View the GPU driver status



What's next

If you want to run the OpenGL program, you must first purchase the licenses and install the GRID drivers. For information about the installation procedure, see the official NVIDIA documentation.

2.1.4.16. Install the CUDA and GPU drivers for a Windows instance

You must install a GPU driver on GPU instances to use the GPU. If the image you use does not contain a pre-installed GPU driver, you must manually install the CUDA and GPU drivers for the instance.

Prerequisites

- If your instance cannot connect to the Internet, the installation file cannot be downloaded. You can install an FTP client on the instance to transfer the installation file to the instance.
- To compile CUDA programs, first install a Windows compiling environment, such as Visual Studio 2015. If you do not need to compile CUDA programs, ignore it.

Procedure

1. Download the CUDA Toolkit.

i. Access the official CUDA download page. Choose the version based on the GPU application requirements for CUDA.

   This example uses CUDA Toolkit 9.2.

ii. Choose a platform based on your operating system. Set **Installer Type** to **exe (local)** and click **Download**.

   NVIDIA drivers are already included in the CUDA Toolkit.

2. Copy the downloaded *cuda_9.2.148_windows.exe* file to the instance. *cuda_9.2.148_windows.exe* is used as an example. Replace it with the actual file name.
3. Double-click *cuda_9.2.148_windows.exe* and follow the installation wizard to install the CUDA driver. *cuda_9.2.148_windows.exe* is used as an example. Replace it with the actual file name.
   The installation takes about 10 to 20 minutes. When `Installed: - Nsight Monitor and HUD Launcher` is displayed, the driver is installed.
4. Press *Win + R* and enter **devmgmt.msc**.
   The NVIDIA device is displayed in **Display Adapter**.
5. Press *Win + R*, enter **cmd**, and run the **"C:\Program Files\NVIDIA Corporation\NVSMI\nvidia-smi"** command.
   If the output displays the details of the GPU driver, the driver is running properly.

What's next

If you want to run the OpenGL and DirectX programs, you must first purchase the licenses and install the GRID drivers. For information about the installation procedure, see the official NVIDIA documentation.

2.1.5. Disks

2.1.5.1. Create a disk

You can create standalone data disks and then attach them to ECS instances to increase the storage space of the instances. This topic describes how to create an empty data disk. You cannot create standalone system disks.

Context

We recommend that you determine the number and sizes of data disks before you create them. The following limits apply to data disks:

- A maximum of 16 data disks can be attached to an instance. Disks and Shared Block Storage devices share this quota.
- Each Shared Block Storage device can be attached to up to four ECS instances at the same time.
- Each ultra disk, standard SSD, Ultra Shared Block Storage device, or SSD Shared Block Storage device can have a maximum capacity of 32 TiB.
- Disks cannot be combined in ECS. They are independent of each other. You cannot combine multiple disks into one by formatting them.

We recommend that you do not use Logical Volume Manager (LVM) to create logical volumes across multiple disks, because a snapshot can back up data only of a single disk. If you create a logical volume across several disks, data discrepancies may occur when you restore these disks.

Procedure

1.
2.
3.
4. Click **Create Disk**.
5. Configure the parameters listed in the following table to create a disk.

Type	Parameter	Required	Description
Region	Organization	Yes	Select an organization in which to create the disk.
	Resource Set	Yes	Select a resource set in which to create the disk.
	Region	Yes	Select a region in which to create the disk.
	Zone	Yes	Select a zone in which to create the disk.
Basic Settings	Name	Yes	Enter a name for the disk. The name must be 2 to 128 characters in length. It must start with a letter and cannot start with http:// or https://. It can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,).
	Specifications	Yes	Select a disk category and specify the disk size. Valid values for the disk category: ○ **Standard SSD**: standard SSD ○ **Ultra Disk**: ultra disk ○ **Shared SSD**: SSD Shared Block Storage device ○ **Shared Ultra Disk**: Ultra Shared Block Storage device The disk size must range from 20 GiB to 32,768 GiB.
	Encrypted	No	Specify whether to encrypt the disk.
	Encryption Method	No	Select an encryption algorithm. This parameter is required when you set **Encrypted** to **Yes**. Valid values: ○ **AES256** ○ **SM4**
	Encryption Key	No	Select a key to be used to encrypt the disk. This parameter is required when you set **Encrypted** to **Yes**. **Note** If no key is available, create a key in KMS.
	Use Snapshot	No	Specify whether to create the disk from a snapshot. If you select **Yes**, you must specify a snapshot. The size of the created disk depends on the size of the specified snapshot. ○ If the disk size that you specify is greater than the snapshot size, the disk is created with the size you specify. ○ If the disk size that you specify is smaller than the snapshot size, the disk is created with the snapshot size.

6. Click **Submit**.

Result

The created disk is displayed in the disk list and in the **Pending** state.

What's next

After the disk is created, you must attach the disk to an instance and partition and format the disk. For more information, see the following topics:

- Attach a disk
- Format a data disk for a Linux instance
- Format a data disk of a Windows instance

2.1.5.2. Attach a disk

You can attach disks that were created separately to instances as data disks. To attach a disk to an instance, make sure that the disk and the instance are in the same region and zone.

Prerequisites

The disk that you want to attach is in the **Pending** state.

Context

- You do not need to attach data disks that are created together with instances.
- A disk can be attached only to a single instance that is in the same zone and region as the disk.
- You can attach a disk to a single instance at a time.
- Each Shared Block Storage device can be attached to up to four instances at the same time.

Procedure

1.
2.
3.
4. Find the disk that you want to attach and choose **More > Attach** in the **Actions** column.
5. Specify the destination instance and configure the release mode.
   - If you select **Release Disk with Instance**, the disk will be released when its associated instance is deleted.
   - If you do not select **Release Disk with Instance**, the disk will be retained and enter the **Pending** state when its associated instance is deleted.
6. Click **OK**.

2.1.5.3. Partition and format disks

2.1.5.3.1. Format a data disk for a Linux instance

Data disks created separately are not partitioned or formatted. This topic describes how to partition and format a data disk of a Linux instance.

Prerequisites

The disk has been attached to the instance.

Procedure

1. Connect to the instance.
2. Run the **fdisk -l** command to view all data disks attached to the ECS instance.

   If */dev/vdb* is not displayed in the command output, the ECS instance does not have a data disk. Check

whether the data disk is attached to the instance.

```
[root@iZ********eZ ~]# fdisk -l
Disk /dev/vda: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00078f9c
Device Boot      Start         End      Blocks   Id  System
/dev/vda1   *           1        5222    41940992   83  Linux
Disk /dev/vdb: 21.5 GB, 21474836480 bytes
16 heads, 63 sectors/track, 41610 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

3. Create partitions for the data disk.
   i. Run the **fdisk /dev/sdb** command.
   ii. Enter *n* to create a new partition.
   iii. Enter *p* to set the partition as the primary partition.
   iv. Enter a partition number and press the Enter key. In this example, *1* is entered to create Partition 1.
   v. Enter the number of the first available sector. This example uses the default value. This is done by pressing the Enter key. You can also enter a value from 1 to 41610 and press the Enter key.
   vi. Enter the number of the last sector. This example uses the default value. This is done by pressing the Enter key. You can also enter a value from 1 to 11748 and press the Enter key.
   vii. (Optional)Optional. To create multiple partitions, repeat steps b through f until all four primary partitions are created.
   viii. Run the **wq** command to start partitioning.

```
[root@iZ********eZ ~]# fdisk /dev/vdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0x01ac58fe.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-41610, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-41610, default 41610):
Using default value 41610
Command (m for help): wq
The partition table has been altered!
```

4. Run the **fdisk -l** command to view the partitions.

   If */dev/vdb1* is displayed in the command output, new partition *vdb1* is created.

```
[root@iZ********eZ ~]# fdisk -l
Disk /dev/vda: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00078f9c
Device Boot      Start         End      Blocks   Id  System
/dev/vda1   *           1        5222    41940992   83  Linux
Disk /dev/vdb: 21.5 GB, 21474836480 bytes
16 heads, 63 sectors/track, 41610 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x01ac58fe
Device Boot      Start         End      Blocks   Id  System
/dev/vdb1               1       41610    20971408+  83  Linux
```

5. Format the new partition. In this example, the new partition is formatted as ext3 after you run the **mkfs.ext3 /dev/vdb1** command.

   The time required for formatting depends on the disk size. You can also format the new partition to another file system. For example, you can run the **mkfs.ext4 /dev/vdb1** command to format the partition as ext4.

   Compared with ext2, ext3 only adds the log function. Compared with ext3, ext4 improves on some important data structures. ext4 provides better performance and reliability, and more functions.

```
[root@iZ********leZ ~]# mkfs.ext3 /dev/vdb1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
1310720 inodes, 5242852 blocks
262142 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
160 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 25 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

6. Run the `echo '/dev/vdb1 /mnt ext3 defaults 0 0' >> /etc/fstab` command to write the information of the new partition to the /etc/fstab file. You can run the `cat /etc/fstab` command to view the new partition information.

   Ubuntu 12.04 does not support barriers. To write the information of the new partition into the /etc/fstab file, you must run the `echo '/dev/vdb1 /mnt ext3 barrier=0 0 0' >> /etc/fstab` command.

   In this example, the partition information is added to the ext3 file system. You can also modify the ext3 parameter to add the partition information to another type of file system.

To attach the data disk to a specific folder, for example, to store web pages, modify the /mnt part of the preceding command.

```
[root@iZ********eZ ~]# echo '/dev/vdb1 /mnt ext3 defaults 0 0' >> /etc/fstab
[root@iZbp19cdhgdj0aw5r2izleZ ~]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu Aug 14 21:16:42 2014
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=94e4e384-0ace-437f-bc96-057dd64f**** / ext4 defaults,barrier=0 1 1
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
/dev/vdb1 /mnt ext3 defaults 0 0
```

7. Mount the new partitions. Run the **mount -a** command to mount all the partitions listed in */etc/fstab* and run the **df -h** command to view the result.

   If the following information is displayed, the new partitions are mounted and available for use.

```
[root@iZ********eZ ~]# mount -a
[root@iZ********eZ ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1        40G  5.6G   32G  15% /
tmpfs           499M     0  499M   0% /dev/shm
/dev/vdb1        20G  173M   19G   1% /mnt
```

2.1.5.3.2. Format a data disk of a Windows instance

Data disks created separately are not partitioned or formatted. This topic describes how to partition and format a data disk of a Windows instance. This example uses Windows Server 2008.

Prerequisites

The disk has been attached to an instance.

Procedure

1. In the lower-left corner of the screen, click the **Server Manager** icon.
2. In the left-side navigation pane of the Server Manager window, choose **Storage > Disk Management**.
3. Right-click an empty partition and select **New Simple Volume** from the shortcut menu.

   If the disk status is **Offline**, change it to **Online**.
4. Click **Next**.
5. Set the size of the simple volume, which is the partition size. Then click **Next**.

   The default value is the maximum value of the disk space. You can specify the partition size as needed.
6. Specify the drive letter and then click **Next**.
7. Specify the formatting options and then click **Next**.

   We recommend that you format the partition with the default settings provided by the wizard.
8. When the wizard prompts that the partition has been completed, click **Finish** to close the wizard.

2.1.5.4. View disks

You can view the list of disks and the details of individual disks.

Procedure

1.
2.
3.
4. Select a filter option from the drop-down list, enter the relevant information in the search box, and click **search**.

   You can select multiple filter options to narrow down search results.

Filter option	Description
Disk Name	Enter a disk name to search for the disk.
Disk ID	Enter a disk ID to search for the disk.
Instance ID	Enter an instance ID to search for disks that are attached to the instance.
Disk Status	Select a disk status to search for disks in that status. Valid values: ○ **Running** ○ **Pending** ○ **Attaching** ○ **Detaching** ○ **Creating** ○ **Deleting** ○ **Deleted** **Note** Deleted disks are no longer displayed in the disk list. ○ **Initializing** ○ **All Statuses**
Disk Properties	Select a disk type to search for disks of that type. Valid values: ○ **All** ○ **System Disk** ○ **Data Disk**
Policy ID	Enter the ID of an automatic snapshot policy to search for disks that use the policy.
Encryption Key ID	Enter the ID of an encryption key to search for disks that are encrypted by using the key.

5. In the **Disk ID/Name** column, click a disk ID to go to the Disk Details page of the disk.

   The properties and mount information of the disk are displayed on the Disk Details page.

2.1.5.5. Restore a disk

If you have created snapshots for a disk, you can use one of the snapshots to restore the disk to the state when the snapshot was created. The disk restore operation is irreversible. After the disk is restored, the data stored on the disk before the restore operation is performed cannot be recovered. Exercise caution when you perform this operation.

Prerequisites

- Snapshots are created for the disk that you want to restore.
- The disk is not released.
- The instance to which the disk is attached is in the **Stopped** state.

Procedure

1.
2.
3.
4. Select a filtering option from the drop-down list, enter the relevant information in the search bar, and click **Search**.

   You can select multiple filtering options to narrow down the search results.

Filtering option	Description
Snapshot Name	Enter a snapshot name to search for the snapshot.
Snapshot ID	Enter a snapshot ID to search for the snapshot.
Instance ID	Enter an instance ID to search for the snapshots related to the instance.
Disk ID	Enter a disk ID to search for the snapshots related to the disk.
Snapshot Type	Select a snapshot type to search for the snapshots of that type. Options include: ◦ **All** ◦ **User Snapshots**: manual snapshots. ◦ **Automatic snapshots**: automatic snapshots.
Creation Time	Enter a creation time to search for the snapshots that were created at that time.

5. Find the snapshot that you want to use to restore the specified disk and click **Restore Disk** in the **Actions** column.
6. Click **OK**.

2.1.5.6. Modify the attributes of a disk

You can modify the attributes of created disks, such as the settings of the Release Disk with Instance and Release Automatic Snapshots with Disk options.

Procedure

1.
2.

3.
4. Find the disk whose attributes you want to modify and choose **More > Modify Disk Properties** in the **Actions** column.
5. Modify the Release Mode settings.
   - **Release Disk with Instance**: If this option is selected, the disk will be released when its associated instance is deleted. If this option is not selected, the disk will be retained and enter the **Pending** state when its associated instance is deleted.
   - **Release Automatic Snapshots with Disk**: If this option is selected, the automatic snapshots of the disk will be released when the disk is deleted. If this option is not selected, the automatic snapshots will be retained when the disk is deleted.
6. Click **OK**.

2.1.5.7. Modify the description of a disk

You can modify the names and descriptions of created disks.

Procedure

1.
2.
3.
4. Find the disk that you want to modify and choose **More > Modify Disk Description** in the **Actions** column.
5. Modify the name and description of the disk.

   The name must be 2 to 128 characters in length and start with a letter. It can contain letters, digits, periods (.), underscores (_), colons (:), and hyphens (-).

   The description must be 2 to 256 characters in length and cannot start with http:// or https://.
6. Click **OK**.

2.1.5.8. Expand a disk

You can expand system disks or data disks online. After a disk is expanded, you do not need to restart the instance to which the disk is attached for the new disk capacity to take effect.

Prerequisites

- To avoid data loss, we recommend that you create a snapshot to back up disk data before you expand a disk. For more information, see Create a snapshot.
- No snapshot is being created for the disk to be expanded.
- The following requirements are met:
  - If the disk is a system disk, the associated instance is in the **Running** state.
  - If the disk is a data disk, one of the following requirements is met:
    - The disk is in the **Pending** state.
    - If the disk is attached to an instance, the instance is in the **Running** state.
  - If the disk is a Shared Block Storage device, it is in the **Pending** state.

Context

The following limits apply when you expand a disk.

Limit	Description
Disk category	• Ultra disks and standard SSDs can be expanded. • SSD Shared Block Storage and Ultra Shared Block Storage devices can be expanded.
Operating system	The system disks of Windows Server 2003 instances cannot be expanded.
Partitioning mode	Data disks that use the MBR partitioning scheme cannot be expanded to larger than 2 TiB in size. To expand an MBR data disk to larger than 2 TiB in size, we recommend that you create and attach a new data disk larger than 2 TiB in size, use the GPT partitioning scheme to partition this new data disk, and then copy data from the original MBR data disk to the new GPT data disk.
File system	For Windows instances, only disks that use NTFS file systems can be expanded.
Maximum capacity	• Ultra disk and standard SSD: 32,768 GiB • SSD Shared Block Storage device and Ultra Shared Block Storage device: 32,768 GiB
Related operations	• When you expand disks, only the capacity of the disks is expanded. The sizes of partitions and file systems do not change. You must manually re-allocate the storage space on a disk after the disk is expanded. • You cannot shrink an expanded disk by any means, such as by rolling it back.

Procedure

1.
2.
3.
4. Find the disk that you want to expand and choose **More > Expand Disk** in the **Actions** column.
5. In the Expand Disk dialog box, specify a new capacity for the disk.

   The new capacity must be greater than the current capacity.
6. Click **OK**.

Result

When you expand disks, only the capacity of the disks is expanded. The sizes of partitions and file systems do not change. You must manually re-allocate the storage space on a disk after the disk is expanded.

2.1.5.9. Encrypt a disk

2.1.5.9.1. Encrypt a system disk

In the scenarios that require data security and regulatory compliance, you can encrypt disks to secure your data stored on the disks. To encrypt system disks, you can encrypt custom images and then use the encrypted custom images to create instances. The system disks of the created instances are automatically encrypted.

Context

You can encrypt system disks only by encrypting custom images.

Step 1: Create a custom image from an instance

1.
2.

3.
4. Find the snapshot from which you want to create a custom image and click **Create Custom Image** in the **Actions** column.
5. Configure the parameters listed in the following table to create a custom image.

Parameter	Description
Custom Image Name	Enter a name for the custom image. The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), hyphens (-), and colons (:). It must start with a letter.
Sharing Scope	Select the scope in which to share the custom image. ○ **Current Organization and Subordinate Organizations** ○ **Current Resource Set** ○ **Current Organization**
Description	Enter a description for the custom image. The description must be 2 to 256 characters in length and cannot start with http:// or https://.

6. Click **OK**.

Step 2: Encrypt the custom image

1.
2.
3.
4. Click the **Custom Images** tab.
5. Find the custom image that you want to encrypt and click **Encrypt Image** in the **Actions** column.
6. In the **Encrypt Image** dialog box, configure the parameters listed in the following table.

Parameter	Description
Image ID	The system automatically obtains the ID of the image to be encrypted. You do not need to configure this parameter.
Custom Image Name	Enter a name for the new encrypted custom image. The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-). It must start with a letter.
Description	Enter a description for the new encrypted custom image. The description must be 2 to 256 characters in length and cannot start with http:// or https://.

7. Click **OK**.

Step 3: Use the encrypted custom image to create an instance

1.

2.
3.
4. Click **Create Instance**.
5. Configure the parameters for the instance.

   For more information about how to configure these parameters, see Create an instance

   In the **Image** section, set Image Type to **Custom Image**. Then, select the image that you encrypted from the **Custom Image** drop-down list.
6. Click **Submit**.

Result

After the instance is created, you can click its ID to go to the **Instance Details** page. Then, you can click the **Disks** tab and check the value in the **Disk Encryption** column corresponding to the system disk. If the value is **Yes**, the system disk is encrypted.

2.1.5.9.2. Encrypt a data disk

In the scenarios that require data security and regulatory compliance, you can encrypt disks to secure your data stored on the disks. After a data disk is created, you cannot change its encryption state. If you want to encrypt a data disk, enable encryption for the disk when you create it.

Context

We recommend that you determine the number and sizes of data disks before you create them. The following limits apply to data disks:

- A maximum of 16 data disks can be attached to an instance. Disks and Shared Block Storage devices share this quota.
- Each Shared Block Storage device can be attached to up to four ECS instances at the same time.
- Each ultra disk, standard SSD, Ultra Shared Block Storage device, or SSD Shared Block Storage device can have a maximum capacity of 32 TiB.
- Disks cannot be combined in ECS. They are independent of each other. You cannot combine multiple disks into one by formatting them.

We recommend that you do not use Logical Volume Manager (LVM) to create logical volumes across multiple disks, because a snapshot can back up data only of a single disk. If you create a logical volume across several disks, data discrepancies may occur when you restore these disks.

Procedure

1.
2.
3.
4. Click **Create Disk**.
5. On the **Create Disk** page, configure the parameters for the disk.

   When you encrypt the disk, take note of the following parameters:

   - **Encrypted**: Select **Yes**.
   - **Encryption Method**: Select an encryption algorithm.
     - **AES256**: indicates the AES256 encryption algorithm.
   - **Encryption Key**: Select an encryption key.

   For information about how to configure the other parameters to create a disk, see Create a disk.

6. Click **Submit**.

2.1.5.10. Re-initialize a disk

You can re-initialize disks to restore them to their initial states.

Prerequisites

- The disk to be re-initialized is in the **Running** state.
- The instance to which the disk is attached is in the **Stopped** state.
- After a disk is re-initialized, the data stored on the disk is lost and cannot be recovered. Exercise caution when you perform this operation. We recommend that you back up disk data or create snapshots before you re-initialize a disk. For more information, see Create a snapshot.

Context

The result of disk re-initialization depends on the disk type and how the disk was created.

- System disk:
  - The disk is restored to the initial state of the image from which the disk was created.
  - If the corresponding image has been deleted, the disk cannot be re-initialized.
- Data disk:
  - If the disk is empty when created, the disk is restored to an empty disk.
  - If the disk was created from a snapshot, the disk is restored to the state of the snapshot.
  - If the snapshot from which a disk was created has been deleted, the disk cannot be re-initialized.

Procedure

1.
2.
3.
4. Find the data disk that you want to re-initialize and click **Re-initialize** in the **Actions** column.
5. In the Re-initialize Disk dialog box, perform operations based on the disk type.
   - For a system disk, enter and confirm a new instance logon password. Select or clear Instance After Re-initializing Disk, and click **OK**.

     The password must be 8 to 30 characters in length, and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The supported special characters include `( ) ' ~ ! @#$%^&*-_+=|{}[]:;'<>,.? /`
   - For a data disk, click **OK**.

Result

When the disk is being re-initialized, the disk is in the **Initializing** state. After the disk is re-initialized, it enters the **Running** state.

2.1.5.11. Detach a data disk

You can detach data disks but cannot detach system disks.

Prerequisites

- Before you can detach a data disk from a Windows instance, you must bring the data disk offline by using Disk Management.

> **Note** To avoid data loss and ensure data integrity, we recommend that you stop read and write operations on the disk before you detach it.

- Before you can detach a data disk from a Linux instance, you must connect to the instance and unmount all partitions on the disk.

  > **Note** If you have added an entry in the */etc/fstab* file for the disk partitions to be automatically mounted on instance startup, you must delete the entry from the */etc/fstab* file before you detach the disk. Otherwise, you are unable to connect to the instance after the instance is restarted.

- The data disk that you want to detach is in the **Running** state.

Procedure

1.
2.
3.
4. Find the data disk that you want to detach and choose **More > Detach** in the **Actions** column.
5. Click **OK**.

2.1.5.12. Release a data disk

You can release data disks that are no longer needed. Released disks cannot be recovered. Exercise caution when you release data disks.

Prerequisites

The data disk to be released is in the **Pending** state. If the data disk is attached to an instance, you must detach the disk from the instance before you can release the disk.

Procedure

1.
2.
3.
4. Find the disk that you want to release and choose **More > Release** from the **Actions** column.
5. Click **OK**.

2.1.6. Images

2.1.6.1. Create a custom image

You can create a custom image and use it to create identical instances or replace the system disks of existing instances. This way, you can quickly obtain multiple instances with the same operating system and data environment.

Create a custom image from a snapshot

You can create a custom image from a system disk snapshot to load the operating system and data environment of the snapshot to the image. Before you perform this operation, make sure that a system disk snapshot is used. You cannot create custom images from data disk snapshots.

1.
2.

3.
4. Find the snapshot from which you want to create a custom image and click **Create Custom Image** in the **Actions** column.
5. Configure the parameters listed in the following table to create a custom image.

Parameter	Description
Custom Image Name	Enter a name for the custom image. The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), hyphens (-), and colons (:). It must start with a letter.
Sharing Scope	Select the scope in which to share the custom image. ○ **Current Organization and Subordinate Organizations** ○ **Current Resource Set** ○ **Current Organization**
Description	Enter a description for the custom image. The description must be 2 to 256 characters in length and cannot start with http:// or https://.

6. Click **OK**.

Create a custom image from an instance

You can create a custom image from an instance to replicate the data of all disks of the instance, including the system disk and data disks.

> **Note** To avoid data security risks, we recommend that you delete sensitive data from an instance before you use the instance to create a custom image.

When you create a custom image from an instance, a snapshot is automatically generated for each disk on the instance, and all the snapshots constitute a complete custom image.

1.
2.
3.
4. Find the instance from which you want to create a custom image and choose **More > Create Custom Image** in the **Actions** column.
5. Configure the parameters listed in the following table to create a custom image.

Parameter	Description
Custom Image Name	Enter a name for the custom image. The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), hyphens (-), and colons (:). It must start with a letter.

Parameter	Description
Sharing Scope	Select the scope in which to share the custom image. ○ **Current Organization and Subordinate Organizations** ○ **Current Resource Set** ○ **Current Organization**
Description	Enter a description for the custom image. The description must be 2 to 256 characters in length and cannot start with http:// or https://.

6. Click **OK**.

2.1.6.2. View images

You can view a list of images.

Procedure

1.
2.
3.
4. Select a tab based on the type of images that you want to view.

   You can select the **Custom Images** or **Public Images** tab.
5. Select a filter option from the drop-down list, enter the relevant information in the search bar, and click **Search**.

   You can select multiple filter options to narrow down search results.

Filter option	Description
Image Name	Enter an image name to search for the image.
Image ID	Enter an image ID to search for the image.
Snapshot ID	Enter a snapshot ID to search for images associated with the snapshot. This option is not available for public images.

2.1.6.3. View instances related to an image

You can view the instances that use a specified image.

Procedure

1.
2.
3.
4. Select a tab based on the type of the image that you want to view.

   You can select the **Custom Images** or **Public Images** tab.
5. Find the image and click **Related Instances** in the **Actions** column.

Result

The Instances page appears and shows the instances that use the image. You can perform operations on these instances, such as updating the image.

2.1.6.4. Modify the description of a custom image

You can modify the descriptions of created custom images.

Procedure

1.
2.
3.
4. Find the custom image that you want to modify and click **Modify Description** in the **Actions** column.
5. In the Modify Description dialog box, modify the image description in the Basic Settings field.

   The description must be 2 to 256 characters in length and cannot start with http:// or https://.
6. Click **OK**.

2.1.6.5. Share a custom image

You can share a custom image that you created to the organizations that you manage. Then, the organizations can use the shared image to quickly create multiple identical instances.

Context

Only custom images can be shared. Shared images do not count towards the image quotas of the organizations to which the images are shared.

The organizations to which images are shared can use the shared images to create instances or replace the system disks of existing instances.

You can delete shared images. After a shared image is deleted, the image is no longer visible to the organizations to which the image was shared, and the system disks of the instances created from the image can no longer be re-initialized.

Procedure

1.
2.
3.
4. Find the image that you want to share and click **Share Image** in the **Actions** column.
5. Configure the Sharing Scope parameter and click **OK**.

   Valid values of Sharing Scope:
   - **Current Organization and Subordinate Organizations**
   - **Current Resource Set**
   - **Current Organization**

2.1.6.6. Encrypt a custom image

This topic describes how to encrypt a custom image to generate a new identical encrypted custom image.

Prerequisites

The custom image that you want to encrypt is in the Available (Available) state.

Context

To meet the requirements for data security compliance, you can use encrypted custom images to create instances. The system disks of the created instances are automatically encrypted.

Procedure

1.
2.
3.
4. Click the **Custom Images** tab.
5. Find the custom image that you want to encrypt and click **Encrypt Image** in the **Actions** column.
6. In the **Encrypt Image** dialog box, configure the parameters listed in the following table.

Parameter	Description
Image ID	The system automatically obtains the ID of the image to be encrypted. You do not need to configure this parameter.
Custom Image Name	Enter a name for the new encrypted custom image. The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-). It must start with a letter.
Description	Enter a description for the new encrypted custom image. The description must be 2 to 256 characters in length and cannot start with http:// or https://.

7. Click **OK**.

Result

After you encrypt the custom image, a new identical encrypted custom image is generated and displayed on the Custom Images tab.

2.1.6.7. Import custom images

2.1.6.7.1. Limits on importing images

This topic describes the limits on importing images. You must understand the limits to ensure that the imported images are available and make the import operation more efficient.

When you import images, take note of the limits described in the following sections:

- Linux images
- Windows images

Linux images

When you import Linux images, take note of the following limits:

- Multiple network interfaces are not supported.
- IPv6 addresses are not supported.
- Each password must be 8 to 30 characters in length. It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.

- Firewalls must be disabled. By default, port 22 is enabled.
- Each Linux system disk must range from 40 GiB to 500 GiB in size.
- DHCP must be enabled in the images.
- SELinux must be disabled.
- The Kernel-based Virtual Machine (KVM) driver must be installed.
- We recommend that you install cloud-init to configure hostnames, NTP sources, and YUM sources.

Limits

Item	Standard operating system image	Non-standard operating system image
Definition	The supported standard 32-bit and 64-bit operating systems include: • CentOS • Ubuntu • SUSE • openSUSE • RedHat • Debian • CoreOS • Aliyun Linux **Note** Support for standard operating systems may be subject to version changes. You can log on to the ECS console to view the latest supported operating systems.	Non-standard operating systems include: • Operating systems that are not supported by Alibaba Cloud • Standard operating systems that do not meet the requirements on critical system configuration files, basic system environments, or applications To use non-standard operating system images, select Others Linux when you import images. If the images that you import are non-standard operating system images, Alibaba Cloud does not process the instances created from these images. After you create an instance from a non-standard operating system image, you must connect to the instance by using the VNC feature in the ECS console, and then configure the IP address, route, and password of the instance.
Critical system configuration file	• Do not modify `/etc/issue*`. Otherwise, the version of the operating system cannot be identified, which leads to a failure to create the system. • Do not modify `/boot/grub/menu.lst`. Otherwise, the system fails to start. • Do not modify `/etc/fstab`. Otherwise, partitions cannot be loaded, which causes the system to fail to start. • Do not change `/etc/shadow` to read-only. Otherwise, the password file cannot be modified, which leads to a failure to create the system. • Do not modify `/etc/selinux/config` to enable SELinux. Otherwise, the system fails to start.	

Item	Standard operating system image	Non-standard operating system image
Basic system environment	• Do not adjust the system disk partitions. Only disks with a single root partition are supported. • Make sure that the system disk has sufficient free space. • Do not modify critical system files such as `/sbin`, `/bin`, and `/lib*`. • Before you import an image, confirm the integrity of the file system. • Only ext3 and ext4 file systems are supported.	Requirements for standard operating system images are not met.
Application	Do not install `qemu-ga` on a custom image. Otherwise, some of the services that Alibaba Cloud uses may be unavailable.	
Image file format	Only images in the RAW, VHD, or QCOW2 format can be imported. To import images in other formats, use a tool to convert the images to a supported format. We recommend that you import images in the VHD format, which has a smaller transmission footprint.	
Image file size	We recommend that you configure the system disk size based on the virtual disk size rather than the image size. The configured system disk size must be at least 40 GiB.	

Windows images

When you import Windows images, take note of the following limits:

- Each password must be 8 to 30 characters in length. It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.
- Imported Windows images do not provide the Windows Activation Service.
- Firewalls must be disabled. Otherwise, remote logon is not supported. Port 3389 must be enabled.
- Each Windows system disk must range from 40 GiB to 500 GiB in size.

Limits

Item	Description

Item	Description

Operating system version	Alibaba Cloud allows you to import the following 32-bit and 64-bit versions of Windows operating system images:
	• Microsoft Windows Server 2016
	• Microsoft Windows Server 2012, including:
	  ◦ Microsoft Windows Server 2012 R2 (Standard Edition)
	  ◦ Microsoft Windows Server 2012 (Standard Edition and Datacenter Edition)
	• Microsoft Windows Server 2008, including:
	  ◦ Microsoft Windows Server 2008 R2 (Standard Edition, Datacenter Edition, and Enterprise Edition)
	  ◦ Microsoft Windows Server 2008 (Standard Edition, Datacenter Edition, and Enterprise Edition)
	• Microsoft Windows Server 2003, including:
	  ◦ Microsoft Windows Server 2003 R2 (Standard Edition, Datacenter Edition, and Enterprise Edition)
	  ◦ Microsoft Windows Server 2003 (Standard Edition, Datacenter Edition, and Enterprise Edition) or later, including Service Pack 1 (SP1)
	• Microsoft Windows 7, including:
	  ◦ Microsoft Windows 7 (Professional Edition)
	  ◦ Microsoft Windows 7 (Enterprise Edition)
	Note Support for standard operating systems may be subject to version changes. You can log on to the ECS console to view the latest supported operating systems.

Basic system environment	• Multi-partition system disks are supported. • Make sure that the system disk has sufficient free space. • Do not modify critical system files. • Before you import an image, confirm the integrity of the file system. • Disks can be partitioned in the MBR format and formatted to NTFS file systems.
Application	Do not install qemu-ga on an imported image. Otherwise, some of the services that Alibaba Cloud uses may be unavailable.
Supported image file format	• RAW • VHD • QCOW2 We recommend that you configure the system disk size based on the virtual disk size rather than the image size. The configured system disk size must range from 40 GiB to 500 GiB. **Note** We recommend that you import images in the VHD format, which has a smaller transmission footprint.

2.1.6.7.2. Convert the image file format

You can only import image files in the RAW, VHD, and qcow2 formats to ECS. If you want to import images in other formats, you must convert the image into a supported format. This topic describes how to convert the image format in Windows and Linux.

Context

You can use the qemu-img tool to convert an image from VMDK, VDI, VHDX, qcow1, or QED to RAW, VHD, or qcow2, or implement conversion between RAW, VHD, and qcow2.

> **Note** We recommend that you use the qcow2 format if your application environment supports this format.

Windows

1. Download qemu.

   Visit QEMU Binaries for Windows (64 bit) to download the qemu tool. Select a qemu version based on your operating system.

2. Install qemu.

   The installation path in this example is *C:\Program Files\qemu*.

3. Configure the environment variables for qemu.

   i. Choose **Start > Computer**, right-click Computer, and choose **Properties** from the shortcut menu.

   ii. In the left-side navigation pane, click **Advanced System Settings**.

   iii. In the **System Properties** dialog box that appears, click the **Advanced** tab and then click **Environment Variables**.

   iv. In the **Environment Variables** dialog box that appears, find the **Path** variable from the **System variables** section.

   - If the **Path** variable exists, click **Edit**.
   - If the **Path** variable does not exist, click **New**.

   v. Add a system variable value.

   - In the Edit System Variable dialog box that appears, add *C:\Program Files\qemu* to the **Variable value** field, separate different variable values with semicolons (;), and then click **OK**.
   - In the New System Variable dialog box that appears, enter *Path* in the **Variable name** field, enter *C:\Program Files\qemu* in the **Variable value** field, and then click **OK**.

4. Open Command Prompt in Windows and run the `qemu-img --help` command. If a successful response is displayed, the tool is installed.

5. In the Command Prompt window, run the `cd [Directory of the source image file]` command to switch to a new file directory,

   for example, `cd D:\ConvertImage`.

6. In the Command Prompt window, run the `qemu-img convert -f raw -O qcow2 centos.raw centos.qcow2` command to convert the image file format.

   The parameters are described as follows:

   - The `-f` parameter is followed by the source image format.
   - The `-O` parameter (case-sensitive) is followed by the destination image format, source file name, and destination file name.

   After the conversion is complete, the destination file appears in the directory of the source image file.

Linux

1. Install the qemu-img tool.
   - For Ubuntu, run the `apt install qemu-img` command.
   - For CentOS, run the `yum install qemu-img` command.
2. Run the `qemu-img convert -f raw -O qcow2 centos.raw centos.qcow2` command to convert the image file format.

   The parameters are described as follows:
   - The `-f` parameter is followed by the source image format.
   - The `-O` parameter (case-sensitive) is followed by the destination image format, source file name, and destination file name.

2.1.6.7.3. Import an image

After you upload a local image to an OSS bucket, you can import the image as a custom image to ECS.

Prerequisites

- An image is made. It meets the limits and requirements for image import and is in the RAW, VHD, or QCOW2 format. For more information, see Limits on importing custom images and Convert the image file format.
- You are authorized to import images. For more information, see the "RAM" chapter in *Apsara Uni-manager Management Console User Guide*.
- A local image is uploaded to a bucket by using the OSS console or by calling an OSS API operation. For more information, see the "Upload objects" topic in *OSS User Guide* or the "PutObject" topic in *OSS Developer Guide*.

> **Note** Make sure that the bucket resides in the region to which you want to import the image as a custom image.

Procedure

1.
2.
3.
4. Click **Import Image**.
5. Configure the parameters listed in the following table to import the image.

Parameter	Required	Description
Region	Yes	The region to which to import the image as a custom image.
Organization	Yes	The organization in which to use the custom image.
Resource Set	Yes	The resource set to which to assign the custom image.
OSS Bucket Name	Yes	The name of the OSS bucket where the image to be imported is stored.
OSS Object Name	Yes	The URL of the object as which the image to be imported is stored in the OSS bucket. For information about how to obtain the URL of an OSS object, see the "Obtain object URLs" topic in *OSS User Guide*.

Parameter	Required	Description
Image Name	Yes	The name of the custom image. The name must be 2 to 128 characters in length. It must start with a letter and can contain letters, periods (.), underscores (_), and hyphens (-).
Sharing Scope	Yes	The scope in which to share the custom image. ○ **Current Organization and Subordinate Organizations** ○ **Current Resource Set** ○ **Current Organization**
Operating System	Yes	Valid values: **Linux** and **Windows**.
System Disk	Yes	The size of the system disk on an instance. Unit: GiB.
System Architecture	Yes	Valid values: **x86_64** and **i386**.
Platform	Yes	Linux: ○ **CentOS** ○ **Ubuntu** ○ **SUSE** ○ **OpenSUSE** ○ **Debian** ○ **CoreOS** ○ **Aliyun** ○ **Others Linux** ○ **Customized Linux** Windows: ○ **Windows Server 2003** ○ **Windows Server 2008** ○ **Windows Server 2012**
Image Format	Yes	The format of the custom image. Valid values: **RAW**, **VHD**, and **QCOW2**.
Description	No	The description of the custom image.

6. Click **OK**.

Result

You can go to the Images page to view the creation progress of the custom image. For more information, see View images. When 100% is displayed in the Progress column, the custom image is created.

2.1.6.8. Export a custom image

You can export custom images to OSS buckets and then download the images to your local device.

Prerequisites

- OSS is activated and an OSS bucket is created. For more information, see the "Create buckets" topic in *OSS User*

Guide.

- You are authorized to export images. For more information, see the "RAM" chapter in *Apsara Uni-manager Management Console User Guide*.

Context

You can export custom images to the RAW, VHD, or QCOW2 format. After a custom image is exported to an OSS bucket, you can download the image to your local device. For more information, see the "Obtain object URLs" topic in *OSS User Guide*.

Procedure

1.
2.
3.
4. Find the custom image that you want to export and click **Export Image** in the **Actions** column.
5. Configure the parameters listed in the following table.

Parameter	Required	Description
OssBucket	Yes	The name of the OSS bucket where to store the exported image.
Image Type	No	The format in which to export the image. Valid values: **RAW**, **VHD**, and **QCOW2**.
OSS Prefix	No	The prefix of the OSS object to which to export the image. must be 1 to 30 characters in length and can contain digits and letters.

6. Click **OK**.

2.1.6.9. Delete a custom image

You can delete custom images that are no longer needed. Custom images can be deleted but public images cannot.

Procedure

1.
2.
3.
4. Use one of the following methods to delete custom images:
   - To delete a single custom image, find the image and click **Delete Image** in the **Actions** column.
   - To delete one or more custom images at a time, select the images and click **Delete** in the lower-left corner of the image list.
5. Click **OK**.

2.1.7. Snapshots

2.1.7.1. Create a snapshot

You can manually create snapshots for disks to back up disk data.

Prerequisites

- The associated instance of the disk for which you want to create a snapshot is in the **Running** or **Stopped** state.
- The disk is in the **Running** state.

Context

You can retain up to 64 snapshots for each disk.

Snapshots can be used in the following scenarios:

- Restore a disk from one of its snapshots.

  For more information, see Restore a disk.
- Create a custom image

  For more information, see Create a custom image from a snapshot. Data disk snapshots cannot be used to create custom images.
- Create a new data disk from a data disk snapshot

  To create a data disk from a snapshot, set Use Snapshot to Yes and then specify a snapshot on the Create Disk page. For more information, see Create a disk. The created disk size is determined by the size of the specified snapshot and cannot be changed. When you re-initialize a data disk created from a snapshot, the disk is restored to the status of the snapshot.

When you create a snapshot, take note of the following items:

- For each disk, the first snapshot is a full snapshot and subsequent snapshots are incremental snapshots. It takes longer to create the first snapshot. The amount of time it takes to create an incremental snapshot depends on the volume of data that has been changed since the last snapshot. The more data that has been changed, the longer it takes.
- We recommend that you create snapshots during off-peak hours.

Procedure

1.
2.
3.
4. Find the disk for which you want to create a snapshot and click **Create Snapshot** in the **Actions** column.
5. Enter a snapshot name and description and then click **OK**.

   > **Note** The names of manual snapshots cannot start with auto because auto is a prefix reserved for automatic snapshots.

   You can go to the Snapshots page to check the creation progress of the snapshot. For more information, see View snapshots. When 100% is displayed in the Progress column, the snapshot is created.

2.1.7.2. View snapshots

You can view the list of snapshots.

Procedure

1.
2.
3.

4. Select a filter option from the drop-down list, enter the relevant information in the search bar, and click **Search**.

   You can select multiple filter options to narrow down search results.

Filter option	Description
Snapshot Name	Enter a snapshot name to search for the snapshot.
Snapshot ID	Enter a snapshot ID to search for the snapshot.
Instance ID	Enter an instance ID to search for snapshots related to the instance.
Disk ID	Enter a disk ID to search for snapshots related to the disk.
Snapshot Type	Select a snapshot type to search for snapshots of that type. Valid values: ○ **All** ○ **User Snapshots**: manual snapshots ○ **Automatic Snapshots**: automatic snapshots
Creation Time	Enter a time to search for snapshots that were created at that time.

2.1.7.3. Delete a snapshot

You can delete snapshots that are no longer needed. Deleted snapshots cannot be recovered. You cannot delete system disk snapshots that have been used to create custom images.

Procedure

1.
2.
3.
4. Use one of the following methods to delete snapshots:
   - To delete a single snapshot, find the snapshot and click **Delete** in the **Actions** column.
   - To delete one or more snapshots at a time, select the snapshots and click **Delete** in the lower-left corner of the Snapshots page.
5. Click **OK**.

2.1.8. Automatic snapshot policies

2.1.8.1. Create an automatic snapshot policy

Automatic snapshot policies can be applied to system disks and data disks to create periodical snapshots of the disks. You can use automatic snapshot policies to improve data security and tolerance against operation faults.

Context

Automatic snapshot policies can effectively eliminate the following risks associated with manual snapshots:

- When applications such as personal websites or databases deployed on an ECS instance encounter attacks or system vulnerabilities, you may be unable to manually create snapshots. In this case, you can use the latest automatic snapshots to roll back the affected disks to restore your data and reduce loss.
- You can also specify an automatic snapshot policy to create snapshots before regular system maintenance tasks are performed. This eliminates the need to manually create snapshots and ensures that snapshots are

always created before maintenance.

You can retain up to 64 snapshots for each disk. If the maximum number of snapshots for a disk is reached while a new snapshot is being created, the system deletes the oldest automatic snapshot.

Procedure

1.
2.
3.
4. Click **Create Policy**.
5. Configure the parameters listed in the following table to create an automatic snapshot policy.

Parameter	Required	Description
Region	Yes	The ID of the region in which to apply the automatic snapshot policy.
Organization	Yes	The organization in which to apply the automatic snapshot policy.
Resource Set	Yes	The resource set in which to apply the automatic snapshot policy.
Policy Name	Yes	The name of the automatic snapshot policy. The name must be 2 to 128 characters in length and cannot start with a special character or digit. It can contain periods (.), underscores (_), hyphens (-), and colons (:).
Creation Time	Yes	The time of the day at which to create an automatic snapshot. Valid values: 00:00 to 23:00 (the start of each hour). You can select multiple values. **Note** The default time zone for the snapshot policy is UTC+8. You can change the time zone to suit your business requirements. If the time scheduled for creating an automatic snapshot is due while a previous automatic snapshot is being created, the new snapshot creation task is skipped. This may occur when a disk contains a large volume of data. For example, assume that an automatic snapshot policy is applied to a disk that contains a large volume of data, and the policy specifies to create snapshots at 00:00, 01:00, and 02:00. If the system starts to create a snapshot at 00:00 and takes 70 minutes to complete the snapshot creation task, the system skips the automatic snapshot task scheduled for 01:00 and creates the next automatic snapshot at 02:00.
Frequency	Yes	The day of the week when to create automatic snapshots. The valid values range from Monday to Sunday. You can select multiple values.

Parameter	Required	Description
Retention Policy	No	The retention period of the automatic snapshots. The default value is Keep for 30 Days. You can select one of the following options: ○ **Keep for**: You can select this option and then specify the number of days during which to retain the automatic snapshots. Valid values: 1 to 65536. ○ **Always Keep the Snapshots Until the Number of Snapshots Reaches the Upper Limit**: You can select this option to retain the automatic snapshots until the maximum number of snapshots is reached.

6. Click **OK**.

What's next

After the automatic snapshot policy is created, you must apply it to a disk for snapshots to be automatically created. For more information, see Configure an automatic snapshot policy for multiple disks.

2.1.8.2. View automatic snapshot policies

You can view the list of automatic snapshot policies.

Procedure

1.
2.
3.
4. View the list of automatic snapshot policies.

2.1.8.3. Modify an automatic snapshot policy

You can modify the attributes of automatic snapshot policies. The attributes of each automatic snapshot policy that can be modified include the name, creation time, frequency, and retention policy.

Procedure

1.
2.
3.
4. Find the automatic snapshot policy that you want to modify and click **Modify Policy** in the **Actions** column.
5. Modify the attributes of the policy.

   Changes made to the retention policy do not affect existing snapshots but take effect only on subsequent snapshots.
6. Click **OK**.

2.1.8.4. Configure an automatic snapshot policy

After you apply an automatic snapshot policy to a disk, snapshots will be created automatically for the disk based on the policy settings. You can cancel an applied automatic snapshot policy at any time.

Context

We recommend that you configure the automatic snapshot policy to create automatic snapshots during off-peak hours. You can also manually create a snapshot for the disk that already has an automatic snapshot policy applied. When an automatic snapshot is being created, you must wait until the snapshot is complete before you can create a manual snapshot. The automatic snapshot is named in the following format: auto_yyyyMMdd_1, such as auto_20140418_1.

Procedure

1.
2.
3.
4. Find the disk and click **Configure Automatic Snapshot Policy** in the **Actions** column.
5. Select a procedure based on the operation you want to perform on the policy.
   - To apply an automatic snapshot policy, turn on **Automatic Snapshot Policy**, select a policy, and then click **OK**.
   - To cancel an automatic snapshot policy, turn off **Automatic Snapshot Policy** and click **OK**.

2.1.8.5. Configure an automatic snapshot policy for multiple disks

After you apply automatic snapshot policies to disks, snapshots are created automatically for the disks based on the policies. You can disable applied automatic snapshot policies at any time.

Context

We recommend that you configure automatic snapshot policies to create automatic snapshots during off-peak hours. You can manually create snapshots for disks that already have automatic snapshot policies applied. When an automatic snapshot is being created for a disk, you must wait for the snapshot to be complete before you can create a manual snapshot for the disk. Each automatic snapshot is named in the following format: auto_yyyyMMdd_1. Example: auto_20140418_1.

Procedure

1.
2.
3.
4. Find the automatic snapshot policy that you want to configure and click **Apply Policy** in the **Actions** column.
5. Apply the policy to disks or disable the policy for disks.
   - To apply the automatic snapshot policy, select the **Disks Without Policy Applied** tab, select one or more disks, and then click **Apply Policy** in the lower part of the tab.
   - To disable the automatic snapshot policy, select the **Disks With Policy Applied** tab, select one or more disks, and then click **Disable Policy** in the lower part of the tab.

2.1.8.6. Delete an automatic snapshot policy

You can delete automatic snapshot policies that are no longer needed. After you delete an automatic snapshot policy, the policy is automatically canceled for the disks that have it applied.

Procedure

1.

2.
3.
4. Find the automatic snapshot policy that you want to delete and click **Delete Policy** in the **Actions** column.
5. Click **OK**.

2.1.9. Security groups

2.1.9.1. Create a security group

Security groups are an important means for network security isolation. They implement network access control for one or more ECS instances.

Prerequisites

A virtual compute cloud (VPC) is created. For more information, see *VPC User Guide*.

Context

Security groups determine whether the instances in the same account that are deployed within the same VPC and region can communicate with each other. By default, if the instances belong to the same security group, they can communicate with each other over the internal network. If the instances belong to different security groups, you can authorize mutual access between the security groups to allow the instances to communicate with each other over the internal network.

Procedure

1.
2.
3.
4. Click **Create Security Group**.
5. Configure the parameters listed in the following table to create a security group.

Type	Parameter	Required	Description
Region	Organization	Yes	Select an organization in which to create the security group. Make sure that the security group and the VPC belong to the same organization.
	Resource Set	Yes	Select a resource set in which to create the security group. Make sure that the security group and the VPC belong to the same resource set.
	Region	Yes	Select a region in which to create the security group. Make sure that the security group and VPC belong to the same region.
	Zone	Yes	Select a zone in which to create the security group.
Basic Settings	VPC	Yes	Select a VPC in which to create the security group.
	Security Group Name	No	Enter a name for the security group. The name must be 2 to 128 characters in length and start with a letter. It can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,). It cannot start with http:// or https://.

Basic Settings Type	Parameter	Required	Description
	Description	No	Enter a description for the security group. To simplify future management operations, we recommend that you provide an informational description. The description must be 2 to 256 characters in length and start with a letter. It can contain letters, digits, periods (.), underscores (_), hyphens (-), and commas (,). It cannot start with http:// or https://.

6. Click **Submit**.

2.1.9.2. View security groups

You can view the list of security groups.

Procedure

1.
2.
3.
4. Select a filter option from the drop-down list, enter the relevant information in the search bar, and click **Search**.

   You can select multiple filter options to narrow down search results.

Filter option	Description
Security Group ID	Enter a security group ID to search for the security group.
Security Group Name	Enter a security group name to search for the security group.
VPC ID	Enter a VPC ID to search for security groups that belong to the VPC.

2.1.9.3. Modify a security group

You can modify the names and descriptions of created security groups.

Procedure

1.
2.
3.
4. Find the security group that you want to modify and click **Modify** in the **Actions** column.
5. Modify the name and description of the security group.
6. Click **OK**.

2.1.9.4. Add a security group rule

You can use security group rules to control access to and from the ECS instances in a security group over the Internet and the internal network.

Procedure

1.
2.
3.
4. Find the security group to which you want to add a rule and click **Rules** in the **Actions** column.
5. Click **Create Rule**.
6. Configure the parameters listed in the following table to create a security group rule.

Parameter	Required	Description
ENI Type	Yes	Valid value: **Internal Network ENI**. In VPCs, you cannot find public NICs in ECS instances and can add only internal security group rules. However, the added security group rules apply to both the Internet and the internal network.
Direction	Yes	◦ **Outbound**: access from the ECS instances in the current security group to other ECS instances on the internal network or to resources on the Internet. ◦ **Inbound**: access from other ECS instances on the internal network or from resources on the Internet to the ECS instances in the current security group.
Action	Yes	◦ **Allow**: allows access requests on specified ports. ◦ **Deny**: discards requests received on specified ports without returning messages. If two security group rules are different only in the Action parameter, the **Deny** rule takes effect whereas the **Allow** rule is ignored.
Protocol	Yes	◦ **All**: This value can be used in total trust scenarios. ◦ **TCP**: This value can be used to allow or deny traffic on one or several successive ports. ◦ **UDP**: This can be used to allow or deny traffic on one or several successive ports. ◦ **ICMP**: This value can be used when the `ping` command is used to test the status of the network connection between instances. ◦ **ICMPv6**: This value can be used when the `ping6` command is used to test the status of the network connection between instances. ◦ **GRE**: This value can be used for VPN.

Parameter	Required	Description
Port Range	Yes	The port range depends on the protocol type. ◦ When you set Protocol to **ALL**, the value of **-1/-1** is displayed, which indicates all ports. You cannot specify a port range in this case. ◦ When you set Protocol to **TCP**, you can specify a port range in the <start port number>/<end port number> format. Valid port numbers: 1 to 65535. You can set the start port number and end port number to the same value to specify a single port. For example, use 22/22 to specify port 22. ◦ When you set Protocol to **UDP**, you can specify a port range in the <start port number>/<end port number> format. Valid port numbers: 1 to 65535. You can set the start port number and end port number to the same value to specify a single port. For example, use 3389/3389 to specify port 3389. ◦ When you set Protocol to **ICMP**, the value of **-1/-1** is displayed, which indicates all ports. You cannot specify a port range in this case. ◦ When you set Protocol to **ICMPv6**, the value of **-1/-1** is displayed, which indicates all ports. You cannot specify a port range in this case. ◦ When you set Protocol to **GRE**, the value of **-1/-1** is displayed, which indicates all ports. You cannot specify a port range in this case.
Priority	Yes	The priority of the rule. Valid values: 1 to 100. The default value is 1, which indicates the highest priority.
Authorization Type	Yes	◦ **IPv4 Addresses**: IPv4 addresses or CIDR blocks. ◦ **IPv6 Addresses**: IPv6 addresses or CIDR blocks. ◦ **Security Groups**: security groups. This authorization type takes effect only on the internal network. You can select another security group in the current account as the authorization object for the instances in the current security group. This way, you can control the access to or from the ECS instances in that security group over the internal network.
Authorization object	Yes	Authorization objects depend on the authorization type. When you set Authorization Type to **IPv4 Addresses**: ◦ Enter single IPv4 addresses or CIDR blocks. Example: *192.0.2.1* or *192.0.2.0/24*. ◦ You can enter up to 10 authorization objects at a time. Separate multiple objects with commas (,). ◦ If you enter *0.0.0.0/0*, all IPv4 addresses are allowed or denied based on the Action parameter. Exercise caution when you specify 0.0.0.0/0. When you set Authorization Type to **IPv6 Addresses**: ◦ Enter single IPv6 addresses or CIDR blocks. Example: *2001:db8:1:1:1:1:1:1* or *2001:db8::/32*. ◦ You can enter up to 10 authorization objects at a time. Separate multiple objects with commas (,). ◦ If you enter *::/0*, all IPv6 addresses are allowed or denied based on the Action parameter. Exercise caution when you specify ::/0. When you set Authorization Type to **Security Groups**, select a security group ID. If the current security group is of the VPC type, the selected security group must be in the same VPC as the current security group.

Parameter	Required	Description
Description	No	The description of the security group rule. To simplify future management operations, we recommend that you provide an informational description. The description must be 2 to 256 characters in length and cannot start with http:// or https://.

7. Click **OK**.

2.1.9.5. Clone a security group rule

You can clone a security group rule to quickly create a similar rule.

Procedure

1.
2.
3.
4. Find the target security group and click **Rules** in the **Actions** column.
5. On the Rules page that appears, click the **Inbound** or **Outbound** tab.
6. Find the target security group rule and click **Clone** in the **Actions** column.
7. In the **Clone Security Group Rule** dialog box, modify the attributes of the security group rule.

   For more information about the attributes of security group rules, see Add a security group rule.
8. Click **OK**.

2.1.9.6. Modify a security group rule

You can modify improper rules in a security group to ensure the security of ECS instances in the security group.

Procedure

1.
2.
3.
4. Find the target security group and click **Rules** in the **Actions** column.
5. On the Rules page that appears, click the **Inbound** or **Outbound** tab.
6. Find the target security group rule and click **Modify** in the **Actions** column.
7. In the **Modify Security Group Rule** dialog box, modify the attributes of the security group rule.

   For more information about the attributes of security group rules, see Add a security group rule.
8. Click **OK**.

2.1.9.7. Export security group rules

You can export security group rules of a security group to a local device for backup.

Procedure

1.
2.
3.

4. Find the target security group and click **Rules** in the **Actions** column.
5. On the Rules page that appears, click the **Inbound** or **Outbound** tab.
6. Click **Export** in the upper-right corner to download and save the rules to a local device.

2.1.9.8. Import security group rules

You can import a local backup file of security group rules into a security group to quickly create or restore security group rules.

Procedure

1.
2.
3.
4. Find the target security group and click **Rules** in the **Actions** column.
5. On the Rules page that appears, click the **Inbound** or **Outbound** tab.
6. Click **Import** in the upper-right corner.
7. In the **Import Rule** dialog box, click **Choose File**.
8. Select the target local backup file of security group rules and click Open. Then, click **OK**.

   The local backup file must be in the CSV format. You can download a template file from the **Import Rule** dialog box.

2.1.9.9. Add an instance to a security group

You can add an existing instance to a security group in the same region. After the instance is added, the rules of the security group automatically apply to the instance.

Procedure

1.
2.
3.
4. Find the security group to which you want to add an instance and click **Manage Instances** in the **Actions** column.
5. Click **Add Instance**.
6. Select an instance and click **OK**.

2.1.9.10. Remove instances from a security group

You can remove instances from security groups, but each of the instances must always belong to at least one security group.

Prerequisites

The instances to be removed belong to two or more security groups.

Context

After an instance is removed from a security group, the instance is isolated from the other instances in the security group. We recommend that you perform all tests in advance to ensure that services can continue to run properly after you remove the instance from the security group.

Procedure

1.
2.
3.
4. Find the security group from which you want to remove instances and click **Manage Instances** in the **Actions** column.
5. On the Instances page, select one or more instances and click **Remove** in the lower-left corner.
6. Click **OK**.

2.1.9.11. Delete a security group

You can delete security groups that are no longer needed.

Prerequisites

No instances exist in the security group that you want to delete.

Procedure

1.
2.
3.
4. Use one of the following methods to delete security groups:
   - To delete a single security group, find the security group and click **Delete** in the **Actions** column.
   - To delete one or more security groups at a time, select the security groups and click **Delete** in the lower-left corner of the Security Groups page.
5. Click **OK**.

2.1.10. Elastic Network Interfaces

2.1.10.1. Create an ENI

You can bind elastic network interfaces (ENIs) to instances to create high-availability clusters and implement fine-grained network management. You can also unbind an ENI from an instance and then bind the ENI to another instance to implement a low-cost failover solution.

Prerequisites

- A virtual private cloud (VPC) and a VSwitch are created. For more information, see Create a VPC and Create a VSwitch in *Apsara Stack VPC User Guide* .
- A security group is available in the VPC. If no security group is available in the VPC, create a security group. For more information, see Create a security group.

Context

ENIs are classified into primary and secondary ENIs.

A primary ENI is created by default when an instance is created in a VPC. This primary ENI has the same lifecycle as the instance and cannot be unbound from the instance.

ENIs created separately are secondary ENIs. You can bind secondary ENIs to or unbind them from instances. This topic describes how to create a secondary ENI.

Procedure

1.
2.
3.
4. Click **Create ENI**.
5. Configure parameters listed in the following table to create an ENI.

Section	Parameter	Required	Description
Region	Organization	Yes	The organization in which to create the ENI.
	Resource Set	Yes	The resource set in which to create the ENI.
	Region	Yes	The region in which to create the ENI.
	Zone	Yes	The zone in which to create the ENI.
	VPC	Yes	The VPC in which to create the ENI. The secondary ENI can be bound only to an instance in the same VPC. **Note** After the ENI is created, you cannot change its VPC.
	VSwitch	Yes	The VSwitch to be associated with the ENI. The secondary ENI can be bound only to an instance in the same VPC. Select a VSwitch that is in the same zone as the instance to which the ENI will be bound. The VSwitch of the ENI can be different from that of the instance. **Note** After an ENI is created, you cannot change its VSwitch.
	Security Group	Yes	The security group in which to create the ENI within the specified VPC. The rules of the security group automatically apply to the ENI.

Basic Settings Section	Parameter	Required	Description
	ENI Name	Yes	The name of the ENI. The name must be 2 to 128 characters in length. It must start with a letter and cannot start with http:// or https://. It can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,).
	Description	No	The description of the ENI. We recommend that you provide an informational description to simplify future management operations. The description must be 2 to 256 characters in length. It must start with a letter and cannot start with http:// or https://. It can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,).
	Primary Private IP	No	The primary private IPv4 address of the ENI. The IPv4 address must be within the CIDR block of the specified VSwitch. If you do not specify a primary private IP address, the system automatically assigns a private IP address to the ENI.

6. Click **Submit**.

Result

The created ENI is displayed on the ENIs page and is in the **Available** state.

2.1.10.2. View ENIs

You can view the list of ENIs.

Procedure

1.
2.
3.
4. Select a filter option from the drop-down list, enter the relevant information in the search bar, and click **Search**.

You can select multiple filter options to narrow down search results.

Filter option	Description
ENI Name	Enter an ENI name to search for the ENI.
ENI ID	Enter an ENI ID to search for the ENI.
VSwitch ID	Enter a vSwitch ID to search for ENIs that are associated with the vSwitch.
Security Group ID	Enter a security group ID to search for ENIs that belong to the security group.
Instance ID	Enter an instance ID to search for ENIs that are bound to the instance.

2.1.10.3. Modify a secondary ENI

You can modify the attributes of a secondary elastic network interface (ENI), including the name, security group, and description.

Prerequisites

The secondary ENI is in the **Available** state.

Procedure

1.
2.
3.
4. Find the secondary ENI and click **Modify** in the **Actions** column.
5. In the Modify ENI dialog box that appears, modify the name, security group, and description of the ENI.
6. Click **OK**.

2.1.10.4. Bind a secondary ENI to an instance

You can bind a secondary elastic network interface (ENI) to an instance. After the ENI is bound to the instance, the instance can process the traffic on the ENI.

Prerequisites

- The secondary ENI is in the **Available** state.
- The instance to which you want to bind the secondary ENI is in the **Running** or **Stopped** state.
- The instance and the secondary ENI belong to the same VPC.
- The VSwitch with which the secondary ENI is associated is in the same zone as the VSwitch to which the instance is connected. An ENI can be bound only to an instance in the same zone. The VSwitches of the ENI and of the instance can be different but must be in the same zone.

Context

The following limits apply when you bind an ENI to an instance:

- You can manually bind only secondary ENIs. Primary ENIs share the same lifecycle as instances and cannot be manually bound.
- An ENI can only be bound to a single ECS instance. However, an ECS instance can be bound with multiple ENIs.

The maximum number of ENIs that can be bound to an instance depends on the instance type. For more information about the number of ENIs that can be bound to an instance of each instance type, see Instance families in *ECS Product Introduction*.

Procedure

1.
2.
3.
4. Find the target secondary ENI and click **Bind** in the **Actions** column.
5. In the Bind dialog box that appears, select an instance and click **OK**.

Result

In the **Status/Creation Time** column, the status of the secondary ENI changes to **Bound**.

2.1.10.5. Unbind a secondary ENI from an instance

You can unbind a secondary elastic network interface (ENI) from an instance. After the secondary ENI is unbound from the instance, the instance no longer processes the traffic on the ENI.

Prerequisites

- The secondary ENI is in the **Bound** state.
- The instance is in the **Running** or **Stopped** state.

Context

Only secondary ENIs can be unbound. Primary ENIs share the same lifecycle as instances and cannot be unbound.

Procedure

1.
2.
3.
4. Find the secondary ENI and click **Unbind** in the **Actions** column.
5. Click **OK**.

Result

In the **Status/Creation Time** column, the status of the secondary ENI changes to **Available**.

2.1.10.6. Delete a secondary ENI

You can delete a secondary elastic network interface (ENI) that is no longer needed.

Prerequisites

The secondary ENI is in the **Available** state.

Context

You can delete only secondary ENIs. Primary ENIs share the same lifecycle as instances and cannot be deleted.

Procedure

1.
2.

3.
4. Find the secondary ENI and click **Delete** in the **Actions** column.
5. Click **OK**.

2.1.11. Deployment sets

2.1.11.1. Create a deployment set

You can use a deployment set to distribute or aggregate instances involved in your business. You can select hosts, racks, or network switches to improve service availability or network performance.

Procedure

1.
2.
3.
4. Click **Create Deployment Set**.
5. Configure the parameters listed in the following table to create a deployment set.

Type	Parameter	Required	Description
Region	Organization	Yes	The organization in which to create the deployment set.
	Resource Set	Yes	The resource set in which to create the deployment set.
	Region	Yes	The region in which to create the deployment set.
	Zone	Yes	The zone in which to create the deployment set.
	Deployment Domain	Yes	This parameter determines the valid values of Deployment Target. Valid values of Deployment Domain: ○ **Default**: When Default is selected, the valid values of Deployment Target are Host, Rack, and Network Switch. ○ **Switch**: When Switch is selected, the valid values of Deployment Target are Host and Rack.
	Deployment Target	Yes	The basic unit that can be scheduled when you deploy instances. ○ **Host**: Instances are distributed or aggregated at the host level. ○ **Rack**: Instances are distributed or aggregated at the rack level. ○ **Network Switch**: Instances are distributed or aggregated at the network switch level.

Type	Parameter	Required	Description
Basic Settings	Deployment Policy	Yes	The dispersion policies are used to improve service availability to avoid impacts on your business when a host, rack, or switch fails. The aggregation policies are used to improve network performance to minimize the access latency between instances. Valid values: ○ **Loose Dispersion** ○ **Strict Dispersion** ○ **Loose Aggregation** ○ **Strict Aggregation**
	Deployment Set Name	No	The name of the deployment set. The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,). It must start with a letter and cannot start with http:// or https://.
	Description	No	The description of the deployment set. To simplify future management operations, we recommend that you provide an informational description. The description must be 2 to 256 characters in length and can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,). It must start with a letter and cannot start with http:// or https://.

6. Click **Submit**.

2.1.11.2. View deployment sets

You can view the list of deployment sets.

Procedure

1.
2.
3.
4. Select a filter option from the drop-down list, enter the relevant information in the search bar, and then click **Search**.

   You can select multiple filter options to narrow down search results.

Filter option	Description
Deployment Set Name	Enter a deployment set name to search for the deployment set.
Deployment Set ID	Enter a deployment set ID to search for the deployment set.

2.1.11.3. Modify a deployment set

You can modify the name and description of a deployment set.

Procedure

1.
2.
3.
4. Find the deployment set and click **Modify** in the **Actions** column.
5. In the Change Deployment Set dialog box, change the name of the deployment set.
6. Click **OK**.

2.1.11.4. Delete a deployment set

You can delete a deployment set that is no longer needed.

Prerequisites

No instances exist in the deployment set.

Procedure

1.
2.
3.
4. Find the deployment set and click **Delete** in the **Actions** column.
5. Click **OK**.

2.1.12. Install FTP software

2.1.12.1. Overview

File Transfer Protocol (FTP) transfers files between a client and a server by establishing two TCP connections. One is the command link for transferring commands between a client and a server. The other is the data link used to upload or download data. Before uploading files to an instance, you must build an FTP site for the instance.

2.1.12.2. Install and configure vsftp in CentOS

This topic describes how to install and configure vsftp in CentOS to transfer files.

Procedure

1. Install vsftp.

   ```
   yum install vsftpd -y
   ```

2. Add an FTP account and a directory.
   i. Check the location of the *nologin* file,

   which is usually under the */usr/sbin* or */sbin* directory.

   ii. Create an FTP account.

   Run the following commands to create the */alidata/www/wwwroot* directory and specify this directory as the home directory of the account pwftp. You can also customize the account name and directory.

   ```
   mkdir -p /alidata/www/wwwroot
   useradd -d /alidata/www/wwwroot -s /sbin/nologin pwftp
   ```

iii. Modify the account password.

```
passwd pwftp
```

iv. Modify the permissions on the specified directory.

```
chown -R pwftp.pwftp /alidata/www/wwwroot
```

3. Configure vsftp.
   i. Open the vsftp configuration file.

```
vi /etc/vsftpd/vsftpd.conf
```

   ii. Change the value of anonymous_enable from `YES` to `NO`.
   iii. Delete the comment delimiter (`#`) from the following configuration lines:

```
local_enable=YES
        write_enable=YES
        chroot_local_user=YES
```

   iv. Press the Esc key to exit the edit mode, and enter `:wq` to save the modifications and exit.
4. Modify the shell configuration.
   i. Open the shell configuration file.

```
vi /etc/shells
```

   ii. If the file does not contain */usr/sbin/nologin* or */sbin/nologin*, add it to the file.
5. Start vsftp and perform a logon test.
   i. Start vsftp.

```
service vsftpd start
```

   ii. Use the account pwftp to perform an FTP logon test.

   This example uses the directory */alidata/www/wwwroot*.

2.1.12.3. Install vsftp in Ubuntu or Debian

This topic describes how to install and configure vsftp in an instance running Ubuntu or Debian to transfer files.

Procedure

1. Update the software source.

```
apt-get update
```

2. Install vsftp.

```
apt-get install vsftpd -y
```

3. Add an FTP account and a directory.
   i. Check the location of the *nologin* file,

   which is typically under the */usr/sbin* or */sbin* directory.

ii. Create an FTP account.

Run the following commands to create the */alidata/www/wwwroot* directory and specify this directory as the home directory of the account pwftp. You can also customize the account name and directory.

```
mkdir -p /alidata/www/wwwroot
useradd -d /alidata/www/wwwroot -s /sbin/nologin pwftp
```

iii. Modify the account password.

```
passwd pwftp
```

iv. Modify the permissions on the specified directory.

```
chown -R pwftp.pwftp /alidata/www/wwwroot
```

4. Configure vsftp.

i. Open the vsftp configuration file.

```
vi /etc/vsftpd.conf
```

ii. Change the value of anonymous_enable from `YES` to `NO`.

iii. Delete the comment delimiter (`#`) from the following configuration lines:

```
local_enable=YES
        write_enable=YES
        chroot_local_user=YES
        chroot_list_enable=YES
        chroot_list_file=/etc/vsftpd.chroot_list
```

iv. Press the Esc key to exit the edit mode, and enter `:wq` to save the modifications and exit.

v. Open the */etc/vsftpd.chroot_list* file and add the FTP account name to the file. Save the modifications and exit.

You can follow steps a to d to open and save the file.

5. Modify shell configurations.

i. Open the shell configuration file.

```
vi /etc/shells
```

ii. If the file does not contain */usr/sbin/nologin* or */sbin/nologin*, add it to the file.

6. Start vsftp and perform a logon test.

i. Start vsftp.

```
service vsftpd restart
```

ii. Use the account pwftp to perform an FTP logon test.

This example uses the directory */alidata/www/wwwroot*.

2.1.12.4. Build an FTP site in Windows Server 2008

This topic describes how to build an FTP site on an instance running Windows Server 2008.

Prerequisites

You have added the Web Server (IIS) role and installed FTP on an instance.

Procedure

1. Connect to an instance.

2. Choose **Start > Adminstrative Tools > Internet Information Services (IIS) Manager**.
3. Right-click the server name and select **Add FTP Site** from the shortcut menu.
4. Enter an FTP site name and a physical path, and then click **Next**.
5. Set **IP Address** to **All Unassigned** and **SSL** to **No SSL**, and then click **Next**.
6. Set **Authentication** to **Basic**, **Authorization** to **All Users**, and **Permissions** to **Read** and **Write**, and click **Finish**.

Result

Then you can use the administrator account and password to upload and download files through FTP. Make sure that the following conditions are met:

- The port for the FTP site is not in use by other applications, and Windows firewall is not blocking the port.
- The security group of the instance contains a security group rule that allows inbound access to the FTP port.

2.1.12.5. Build an FTP site in Windows Server 2012

This topic describes how to build an FTP site on an instance running Windows Server 2012.

Prerequisites

You have added the Web Server (IIS) role and installed FTP on an instance.

Procedure

1. Connect to an instance.
2. Click the **Server Manager** icon.
3. In the left-side navigation pane, click **IIS**.
4. In the **Server** area, right-click the server name and select **Internet Information Services (IIS) Manager** from the shortcut menu.
5. Right-click the server name and select **Add FTP Site** from the shortcut menu.
6. Enter an FTP site name and a physical path, and then click **Next**.
7. Set **IP Address** to **All Unassigned** and **SSL** to **No SSL**, and then click **Next**.
8. Set **Authentication** to **Basic**, **Authorization** to **All Users**, and **Permissions** to **Read** and **Write**, and click **Finish**.

Result

Then you can use the administrator account and password to upload and download files through FTP. Make sure that the following conditions are met:

- The port for the FTP site is not in use by other applications, and Windows firewall is not blocking the port.
- The security group of the instance contains a security group rule that allows inbound access to the FTP port.

3.Container Service

3.1. User Guide

3.1.1. What is Container Service?

Container Service provides high-performance, scalable, and enterprise-class management service for Kubernetes containerized applications throughout the application lifecycle.

Container Service simplifies the deployment and scaling operations on Kubernetes clusters. Integrated with services such as virtualization, storage, network, and security, Container Service aims to provide the optimal cloud environment for Kubernetes containerized applications. Alibaba Cloud is a Kubernetes Certified Service Provider (KCSP). As one of the first services to participate in the Certified Kubernetes Conformance Program, Container Service provides you with professional support and services.

3.1.2. Planning and preparation

Before you start using Container Service, you need to create cloud resources such as VPC networks, VSwitches, disks, and OSS buckets based on your application requirements.

Before you create a Kubernetes cluster, make the following preparations:

- **Create a VPC network** (optional)

  To create a cluster in an existing VPC network, you must create the VPC network and VSwitches in advance.
- **Create a volume** (optional)

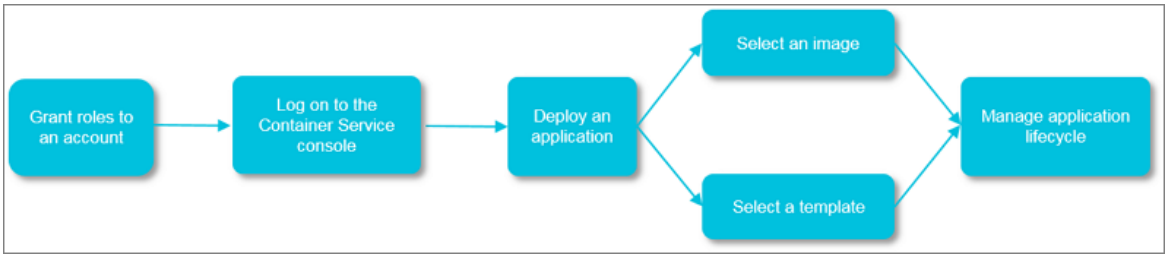
  To create a stateful application with network storage, you must create disks or OSS buckets in advance.

3.1.3. Quick start

3.1.3.1. Procedure

You can perform the following steps to use the Container Service service.

The following diagram shows the procedure to use the Container Service service.



Step 1: Authorize the default role

Authorize the default role of Container Service to perform operations on the resources that belong to the specified organization.

Step 2: Log on to the Container Service console

Log on to the Container Service console. For more information, see Log on to the Container Service console.

Step 3: Create an Container Service cluster

Set the network environment and the number of nodes, and configure node details.

Step 4: Deploy an application by using an image or orchestration template

You can use an existing image or orchestration template, or create a new image or orchestration template.

To create an application that consists of services based on different images, use an orchestration template.

Step 5: Manage the application lifecycle

3.1.3.2. Log on to the Container Service console

You can perform the following steps to log on to the Container Service console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products > Elastic Computing > Container Service for Kubernetes**.
5. Select the required organization and region.
6. Click **ACK** to go to the Container Service console.

3.1.3.3. Log on to the Container Registry console

The following procedure shows how to log on to the Container Registry console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

> **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
>
> - Uppercase or lowercase letters
> - Digits
> - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products > Elastic Computing > Container Registry**.
5. Select an organization and a region.
6. Click **CR** to go to the Container Registry console.

3.1.3.4. Create a Kubernetes cluster

This topic describes how to create a Kubernetes cluster in the Container Service console.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters**. On the Clusters page, click **Create Kubernetes Cluster** in the upper-right corner of the page.
3. On the **Create Cluster** page, set the parameters for the Kubernetes cluster.

Parameter	Description
Cluster Name	Enter a name for the cluster. The name must be 1 to 63 characters in length, and can contain digits, letters, and hyphens (-). **Note** The cluster name must be unique among clusters that belong to the same account.
Region	Select the region in which you want to deploy the Kubernetes cluster.
VPC	You can select a VPC from the drop-down list. - If the selected VPC is associated with a NAT gateway, the cluster uses this NAT gateway. - Otherwise, the system automatically creates a NAT gateway. If you do not want the system to create a NAT gateway, clear **Configure SNAT for VPC**. **Note** If the system does not automatically create a NAT gateway, you must manually configure a NAT gateway to ensure secure communication between the VPC and the Internet. You can also manually configure the Source Network Address Translation (SNAT) feature.

Parameter	Description
vSwitch	Select one or more vSwitches for the Kubernetes cluster. You can select up to three vSwitches that are deployed in different zones.
Kubernetes Version	Check the Kubernetes version.
Container Runtime	Docker and Sandboxed-Container are supported.
Billing Method	Only pay-as-you-go nodes are supported.
Master Configuration	Set the Instance Type and System Disk parameters: ○ Master Node Quantity: Three master instances can be added. ○ Instance Type: You can select one or more instance types. For more information, see the *Instance types* chapter of *ECS User Guide*. ○ System Disk: Standard SSDs and ultra disks are supported. **Note** You can select **Enable Backup** to back up disk data.
Worker Instance	You can choose to create instances or add existing instances to the Kubernetes cluster.
Worker Configuration	If you choose to create worker instances, set the following parameters: ○ Instance Type: You can select one or more instance types. For more information, see the *Instance types* chapter of *ECS User Guide*. ○ Selected Types: The selected instance types. ○ Quantity: Set the number of worker nodes. ○ System Disk: Standard SSDs and ultra disks are supported. **Note** You can select **Enable Backup** to back up disk data. ○ Mount Data Disk: Standard SSDs and ultra disks are supported. **Note** ■ You can select **Encryption** to encrypt disks. ■ You can select **Enable Backup** to back up disk data.

Parameter	Description
Operating system	The CentOS and Aliyun Linux operating systems are supported.
Password	Set the node logon password. **Note** The password must be 8 to 30 characters in length, and contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.
Confirm Password	Confirm the node logon password.
Network plug-in	Flannel and Terway are supported. By default, Flannel is enabled.
Pod CIDR Block and Service CIDR (Optional)	For more information, see the *Network planning* chapter of *VPC User Guide*. **Note** These options are available when you select an existing VPC.
Configure SNAT	Optional. If you clear Configure SNAT for VPC, you must create a NAT gateway or configure SNAT rules.
Access to the Internet	Specify whether to expose the API server with an elastic IP address (EIP). The Kubernetes API server provides multiple HTTP-based RESTful API operations that can be used to create, delete, modify, query, and watch resource objects such as pods and services. ◦ If you select this check box, an EIP is created and associated with an internal-facing Server Load Balancer (SLB) instance. Port 6443 that is used by the API server is opened on master nodes. You can connect to and manage the Kubernetes cluster by using kubeconfig over the Internet. ◦ If you clear this check box, no EIP is created. You can connect to and manage the Kubernetes cluster only by using kubeconfig from within the VPC.
Ingress	Specify whether to install Ingress components. By default, **Install Ingress Component** is installed.
Log Service	If you enable Log Service, you can select an existing project or create a project. If you select **Enable Log Service**, the Log Service plug-in is automatically installed in the Kubernetes cluster. Select **Provide Ingress Dashboard** to support the Ingress access log analysis and monitoring dashboard.
Volume Plug-in	By default, **CSI** is selected.

Parameter	Description
Deletion Protection	If you select this check box, you cannot call API operations or use the Container Service console to delete the Kubernetes cluster.
RDS Whitelist	Add the IP addresses of nodes to the RDS whitelist. **Note** To enable an RDS instance to access the Kubernetes cluster, you must deploy the RDS instance in the same VPC as the Kubernetes cluster.
Node Protection	This check box is selected by default to prevent nodes from being deleted in the Container Service console or by API requests.
Label	Attach labels to the nodes.

4. Advanced settings

Parameter	Description
IP Addresses per Node	The maximum number of IP addresses that can be assigned to a node.
Custom Image	You can create a custom ECS image. All nodes of the cluster are deployed based on this image.
Kube-proxy Mode	**iptables** and **IPVS** are supported. ○ iptables is a kube-proxy mode. It uses iptables rules to conduct service discovery and load balancing. The performance of this mode is restricted by the size of the Kubernetes cluster. This mode is suitable for Kubernetes clusters that run a small number of services. ○ IPVS is a high-performance kube-proxy mode. It uses Linux Virtual Server (LVS) to conduct service discovery and load balancing. This mode is suitable for Kubernetes clusters that run a large number of services. We recommend that you use this mode in scenarios where high-performance load balancing is required.
Custom Node Name	Specify whether to use a custom node name.
Node Port Range	Specify the node port range.
Taints	Add taints to all worker nodes in the Kubernetes cluster.
CPU Policy	Set the CPU policy. ○ none: the default CPU policy. This policy enables the default CPU affinity scheme. ○ static: This policy allows pods with specific resource characteristics on the node to be granted with enhanced CPU affinity and exclusivity.

Parameter	Description
Cluster Domain	The default cluster domain is cluster.local. Custom domains are supported.
Cluster CA	Specify whether to enable the cluster CA certificate.
User Data	Customize the startup behaviors of ECS instances and import data to the ECS instances. The user data can be used to: ○ Run user data scripts during instance startup. ○ Pass user data as common data into an ECS instance for future reference.

5. Click **Create Cluster** in the upper-right corner of the page.
6. On the **Confirm Configuration** page, after all check items are passed, select the terms of service and disclaimer and click **OK** to start deployment.

Result

After the Kubernetes cluster is created, you can find the cluster on the **Clusters** page in the Container Service console.

3.1.3.5. Create an application from an orchestration template

Container Service provides orchestration templates that you can use to create applications quickly. You can also modify the templates based on YAML syntax to customize applications.

Prerequisites

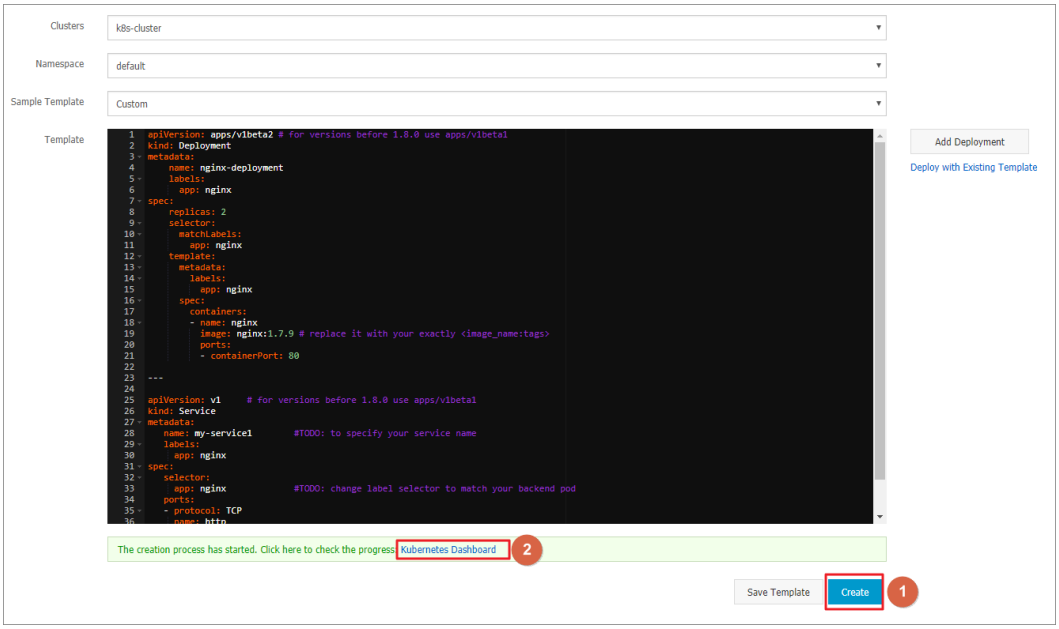
You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.

Context

The following example demonstrates how to create an NGINX application consisting of a deployment and a service. The service is associated with a pod created by the deployment.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Applications > Deployments**. The **Deployments** page appears.
3. In the upper-right corner, click **Create from Template**.
4. Set the parameters and click **Create**.
   - **Cluster**: Select the cluster where the resource objects are to be deployed.
   - **Namespace**: Select the namespace to which the resource objects belong. The default namespace is default. Except for underlying computing resources such as nodes and PVs, most resources are scoped to namespaces.
   - **Sample Template**: Container Service provides YAML templates of various resource types to help you deploy resource objects quickly. You can also create a custom template based on YAML syntax to describe the resource that you want to define.
   - **Add Deployment**: This feature allows you to quickly define a YAML template.
   - **Use Existing Template**: You can import an existing template to the configuration page.



Based on an orchestration template provided by Container Service, the following sample template creates a deployment of an NGINX application.

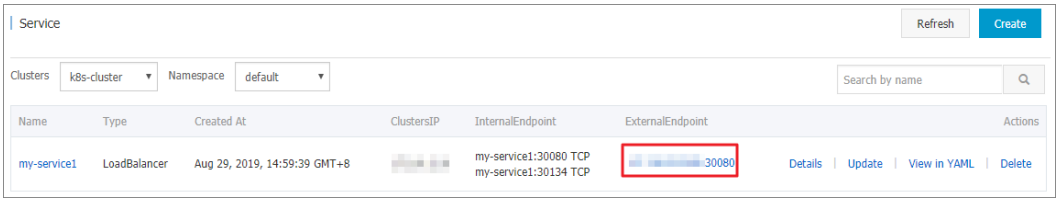
> **Note** Container Service supports YAML syntax. You can use the `---` symbol to separate multiple resource objects. This enables you to create multiple resource objects in a single template.

```
apiVersion: apps/v1beta2 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
    name: nginx-deployment
    labels:
      app: nginx
spec:
    replicas: 2
    selector:
      matchLabels:
        app: nginx
    template:
      metadata:
        labels:
          app: nginx
      spec:
        containers:
        - name: nginx
          image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
          ports:
          - containerPort: 80
---
apiVersion: v1     # for versions before 1.8.0 use apps/v1beta1
kind: Service
metadata:
   name: my-service1        #TODO: to specify your service name
   labels:
     app: nginx
spec:
   selector:
     app: nginx             #TODO: change label selector to match your backend pod
   ports:
   - protocol: TCP
     name: http
     port: 30080            #TODO: choose an unique port on each node to avoid port conflict
     targetPort: 80
   type: LoadBalancer        ## This example changes the service type from NodePort to LoadBalanc
er.
```

5. Click **Create**. A message appears indicating the deployment status.

   In the left-side navigation pane, choose **Ingresses and Load Balancing > Services** to view the newly created service.

6. On Kubernetes Dashboard, verify that a my-service1 service is running and its external endpoint is displayed. Click the address in the **External Endpoint** column.



7. You can visit the NGINX welcome page in the browser.

What's next

You can also choose **Ingresses and Load Balancing > Services** in the left-side navigation pane to view the NGINX service.

3.1.4. Kubernetes clusters

3.1.4.1. Authorizations

3.1.4.1.1. Assign RBAC permissions to a RAM user

This topic describes how to assign role-based access control (RBAC) permissions to Resource Access Management (RAM) users. By default, RBAC policies are enabled for Kubernetes 1.6 and later versions. RBAC policies are important for you to improve the management of clusters. You can use RBAC policies to specify the types of operations that are allowed for specific users based on their roles in an organization.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Authorizations** to go to the Authorizations page.
3. Choose **Select RAM User > RAM Users**, select the required RAM user, and then click **Modify Permissions**.

   > **Note** Before you authorize a RAM user, make sure that the RAM user has been granted the RBAC administrator permissions or the cluster-admin role for the specified cluster.

4. On the **Configure Role-Based Access Control (RBAC)** wizard page, click the plus sign (+) to add permissions on one or all clusters and namespaces, select a predefined role in the Permission field, or click the minus sign (-) for a target role to delete the role, and then click **Next Step**.

   > **Note** You can add permissions of a predefined role and one or more custom roles on a specified cluster or namespace.

   The following table defines the permissions of predefined roles on one or all clusters and namespaces.

   Roles and permissions

Role	RBAC-based permission on one or all clusters
Administrator	Granted the read and write permissions on resources in all namespaces.
O&M Engineer	Granted the read and write permissions on resources in all namespaces and the read-only permissions on nodes, persistent volumes (PVs), namespaces, and quotas.
Developer	Granted the read and write permissions on resources in one or all namespaces.
Restricted User	Granted the read-only permissions on resources in one or all namespaces.

Role	RBAC-based permission on one or all clusters
Custom	Different cluster roles have different permissions. Before you authorize a RAM user, make sure that you are aware of all resource access permissions of the selected cluster roles. This avoids unnecessary permissions granted to the RAM user.

After the authorization, you can use the specified RAM user to log on to the Container Service console and manage the service. For more information, see Log on to the Container Service console.

Custom permissions

Container Service supports predefined roles to grant different permissions. These predefined roles include: administrator, O&M engineer, developer, and restricted user. These predefined roles meet most of your requirements when you manage the service in the console. If you want to customize permissions on clusters, you can use custom roles.

Container Service provides multiple custom roles.

Note The cluster-admin role has permissions of a super administrator of clusters and has permissions on all resources.

You can log on to the master node of the specified cluster and run the following command to view the details of custom permissions.

```
# kubectl get clusterrole
```

```
# kubectl get clusterrole
NAME                                                                   AGE
admin                                                                  13d
alibaba-log-controller                                                 13d
alicloud-disk-controller-runner                                        13d
cluster-admin                                                          13d
cs:admin                                                               13d
edit                                                                   13d
flannel                                                                13d
kube-state-metrics                                                     22h
node-exporter                                                          22h
prometheus-k8s                                                         22h
prometheus-operator                                                    22h
system:aggregate-to-admin                                              13d
....
system:volume-scheduler                                                13d
view                                                                   13d
```

In this example, the cluster-admin role is used. On the command line, run the following command to view the permission details:

```
# kubectl get clusterrole cluster-admin -o yaml
```

Notice After a RAM user is granted the permissions of cluster-admin, for the specified cluster, the RAM user has the same permissions as your Alibaba Cloud account and has all permissions on all resources in the cluster. Proceed with caution.

```
# kubectl get clusterrole cluster-admin  -o yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  creationTimestamp: 2018-10-12T08:31:15Z
  labels:
    kubernetes.io/bootstrapping: rbac-defaults
  name: cluster-admin
  resourceVersion: "57"
  selfLink: /apis/rbac.authorization.k8s.io/v1/clusterroles/cluster-admin
  uid: 2f29f9c5-cdf9-11e8-84bf-00163e0b2f97
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - '*'
- nonResourceURLs:
  - '*'
  verbs:
  - '*'
```

3.1.4.2. Clusters

3.1.4.2.1. Create a Kubernetes cluster

This topic describes how to create a Kubernetes cluster in the Container Service console.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters**. On the Clusters page, click **Create Kubernetes Cluster** in the upper-right corner of the page.
3. On the **Create Cluster** page, set the parameters for the Kubernetes cluster.

Parameter	Description
Cluster Name	Enter a name for the cluster. The name must be 1 to 63 characters in length, and can contain digits, letters, and hyphens (-). **Note** The cluster name must be unique among clusters that belong to the same account.
Region	Select the region in which you want to deploy the Kubernetes cluster.

Parameter	Description
VPC	You can select a VPC from the drop-down list. ◦ If the selected VPC is associated with a NAT gateway, the cluster uses this NAT gateway. ◦ Otherwise, the system automatically creates a NAT gateway. If you do not want the system to create a NAT gateway, clear **Configure SNAT for VPC**. **Note** If the system does not automatically create a NAT gateway, you must manually configure a NAT gateway to ensure secure communication between the VPC and the Internet. You can also manually configure the Source Network Address Translation (SNAT) feature.
vSwitch	Select one or more vSwitches for the Kubernetes cluster. You can select up to three vSwitches that are deployed in different zones.
Kubernetes Version	Check the Kubernetes version.
Container Runtime	Docker and Sandboxed-Container are supported.
Billing Method	Only pay-as-you-go nodes are supported.
Master Configuration	Set the Instance Type and System Disk parameters: ◦ Master Node Quantity: Three master instances can be added. ◦ Instance Type: You can select one or more instance types. For more information, see the *Instance types* chapter of *ECS User Guide*. ◦ System Disk: Standard SSDs and ultra disks are supported. **Note** You can select **Enable Backup** to back up disk data.
Worker Instance	You can choose to create instances or add existing instances to the Kubernetes cluster.

Parameter	Description
Worker Configuration	If you choose to create worker instances, set the following parameters: ◦ Instance Type: You can select one or more instance types. For more information, see the *Instance types* chapter of *ECS User Guide*. ◦ Selected Types: The selected instance types. ◦ Quantity: Set the number of worker nodes. ◦ System Disk: Standard SSDs and ultra disks are supported. **Note** You can select **Enable Backup** to back up disk data. ◦ Mount Data Disk: Standard SSDs and ultra disks are supported. **Note** ■ You can select **Encryption** to encrypt disks. ■ You can select **Enable Backup** to back up disk data.
Operating system	The CentOS and Aliyun Linux operating systems are supported.
Password	Set the node logon password. **Note** The password must be 8 to 30 characters in length, and contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.
Confirm Password	Confirm the node logon password.
Network plug-in	Flannel and Terway are supported. By default, Flannel is enabled.
Pod CIDR Block and Service CIDR (Optional)	For more information, see the *Network planning* chapter of *VPC User Guide*. **Note** These options are available when you select an existing VPC.
Configure SNAT	Optional. If you clear Configure SNAT for VPC, you must create a NAT gateway or configure SNAT rules.

Parameter	Description
Access to the Internet	Specify whether to expose the API server with an elastic IP address (EIP). The Kubernetes API server provides multiple HTTP-based RESTful API operations that can be used to create, delete, modify, query, and watch resource objects such as pods and services. ◦ If you select this check box, an EIP is created and associated with an internal-facing Server Load Balancer (SLB) instance. Port 6443 that is used by the API server is opened on master nodes. You can connect to and manage the Kubernetes cluster by using kubeconfig over the Internet. ◦ If you clear this check box, no EIP is created. You can connect to and manage the Kubernetes cluster only by using kubeconfig from within the VPC.
Ingress	Specify whether to install Ingress components. By default, **Install Ingress Component** is installed.
Log Service	If you enable Log Service, you can select an existing project or create a project. If you select **Enable Log Service**, the Log Service plug-in is automatically installed in the Kubernetes cluster. Select **Provide Ingress Dashboard** to support the Ingress access log analysis and monitoring dashboard.
Volume Plug-in	By default, **CSI** is selected.
Deletion Protection	If you select this check box, you cannot call API operations or use the Container Service console to delete the Kubernetes cluster.
RDS Whitelist	Add the IP addresses of nodes to the RDS whitelist. **Note** To enable an RDS instance to access the Kubernetes cluster, you must deploy the RDS instance in the same VPC as the Kubernetes cluster.
Node Protection	This check box is selected by default to prevent nodes from being deleted in the Container Service console or by API requests.
Label	Attach labels to the nodes.

4. Advanced settings

Parameter	Description
IP Addresses per Node	The maximum number of IP addresses that can be assigned to a node.
Custom Image	You can create a custom ECS image. All nodes of the cluster are deployed based on this image.

Parameter	Description
Kube-proxy Mode	**iptables** and **IPVS** are supported. ◦ iptables is a kube-proxy mode. It uses iptables rules to conduct service discovery and load balancing. The performance of this mode is restricted by the size of the Kubernetes cluster. This mode is suitable for Kubernetes clusters that run a small number of services. ◦ IPVS is a high-performance kube-proxy mode. It uses Linux Virtual Server (LVS) to conduct service discovery and load balancing. This mode is suitable for Kubernetes clusters that run a large number of services. We recommend that you use this mode in scenarios where high-performance load balancing is required.
Custom Node Name	Specify whether to use a custom node name.
Node Port Range	Specify the node port range.
Taints	Add taints to all worker nodes in the Kubernetes cluster.
CPU Policy	Set the CPU policy. ◦ none: the default CPU policy. This policy enables the default CPU affinity scheme. ◦ static: This policy allows pods with specific resource characteristics on the node to be granted with enhanced CPU affinity and exclusivity.
Cluster Domain	The default cluster domain is cluster.local. Custom domains are supported.
Cluster CA	Specify whether to enable the cluster CA certificate.
User Data	Customize the startup behaviors of ECS instances and import data to the ECS instances. The user data can be used to: ◦ Run user data scripts during instance startup. ◦ Pass user data as common data into an ECS instance for future reference.

5. Click **Create Cluster** in the upper-right corner of the page.
6. On the **Confirm Configuration** page, after all check items are passed, select the terms of service and disclaimer and click **OK** to start deployment.

Result

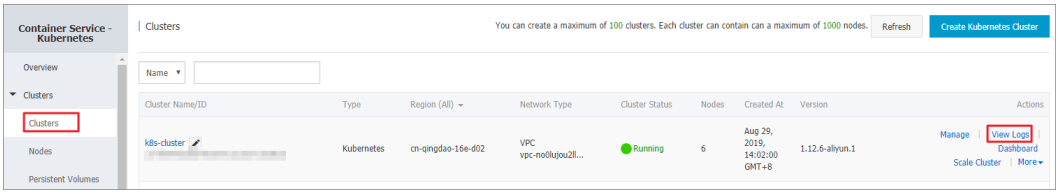
After the Kubernetes cluster is created, you can find the cluster on the **Clusters** page in the Container Service console.

3.1.4.2.2. View cluster logs

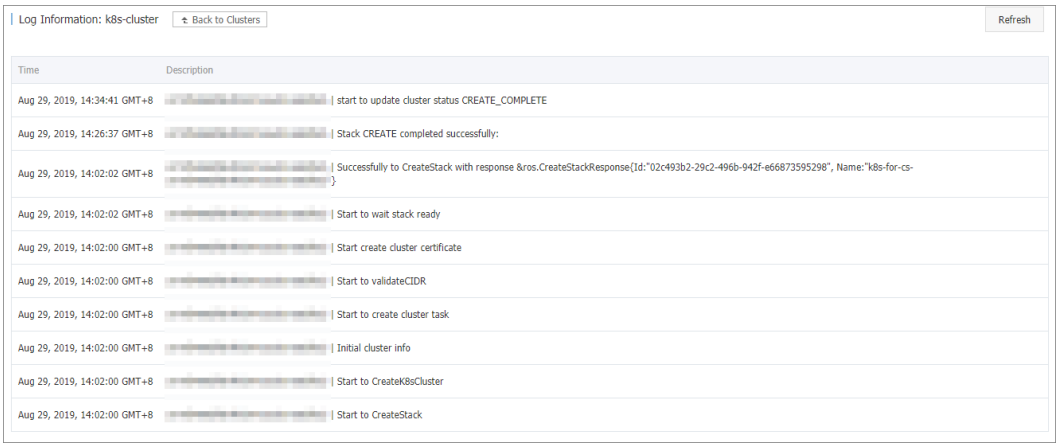
You can view operation logs through the console.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters**. The Clusters page appears.
3. Find the target cluster and click **View Logs** in the Actions column.



You can view operations performed on the cluster.



3.1.4.2.3. Connect to a cluster through kubectl

You can use the Kubernetes command line tool, kubectl, to connect to a Kubernetes cluster from a local computer.

Procedure

1. Download the latest kubectl client from the Kubernetes change log page.
2. Install and set up the kubectl client.

   For more information, see Install and set up kubectl.
3. Configure the cluster credentials.

   You can use the `scp` command to securely copy the master node configuration file from the `/etc/kubernetes/kube.conf` directory of the master VM and paste it to the `$HOME/.kube/config` directory of the local computer, where the `kubectl` credentials are expected to be stored.

   ```
   mkdir $HOME/.kube
   scp root@<master-public-ip>:/etc/kubernetes/kube.conf $HOME/.kube/config
   ```

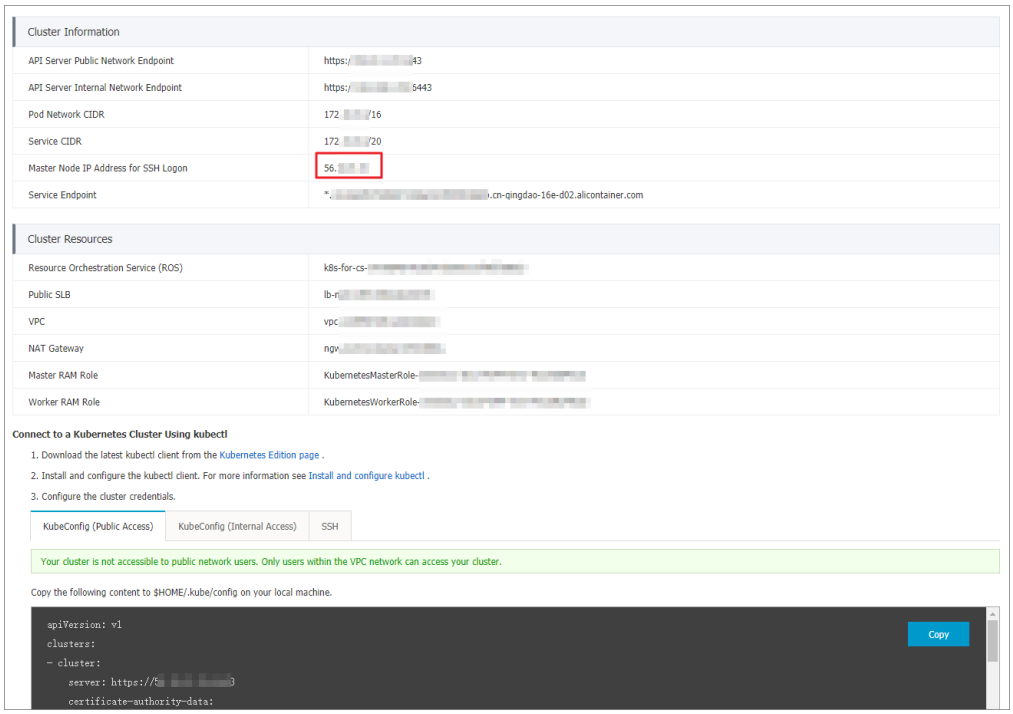
   You can find `master-public-ip` on the cluster details page.

   i. Log on to the Container Service console.

   ii. In the left-side navigation pane, click **Clusters**. The Clusters page appears.

iii. Find the target cluster and click **Manage** in the Actions column.

In the **Cluster Information** section, you can find the master node IP address.



3.1.4.2.4. Connect to a master node by using SSH

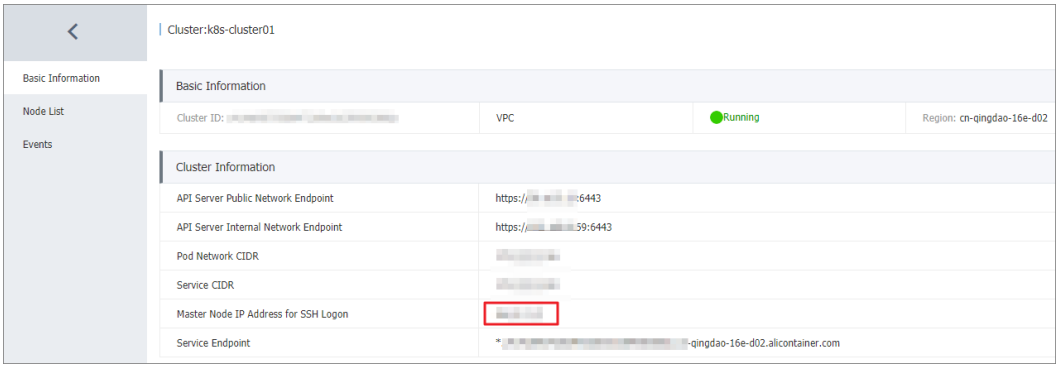
You can access a master node in a cluster by using a Secure Shell (SSH) client.

Prerequisites

- A Kubernetes cluster is created and **Use SSH to Access the Cluster from the Internet** is selected for the cluster. For more information, see Create a Kubernetes cluster.
- The SSH client can connect to the network where the cluster is deployed.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters** > **Clusters** to go to the Clusters page. Find the cluster that you want to manage, and click Manage in the Actions column for the cluster.
3. The Basic Information page appears. In the Cluster Information section, you can find the IP address that is displayed in the **Master Node IP Address for SSH Logon** field.



4. Use SSH to connect to the cluster from an SSH client that has access to the cluster network.
   - If you have a leased line that connects to the cluster network over the Internet, you can use tools such as PuTTY to create an SSH connection.
   - If you have an Elastic Compute Service (ECS) instance that is connected to the Virtual Private Cloud (VPC) network of the cluster, run the following command to create an SSH connection:

     ```
     ssh    root@ssh_ip                    #ssh_ip specifies the IP address of the master node for SSH connection.
     ```

3.1.4.2.5. Expand a Container Service cluster

This topic describes how to scale out the worker nodes of a Container Service cluster in the Container Service console.

Context

You cannot scale out the master nodes in a Container Service cluster.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters**. The Clusters page appears.
3. On the **Clusters** page, find the Container Service cluster that you want to manage and click **Expand** in the **Actions** column.
4. Go to the Expand page and set the required parameters.

   In this example, the number of worker nodes in the Container Service cluster is increased from three to five. The following table describes the required parameters.

Parameter	Description
Cluster Name	By default, the name of the Container Service cluster appears.
Region	The region where the Container Service cluster is deployed.
Container Runtime	By default, the container runtime of the Container Service cluster appears.
VPC	By default, the virtual private cloud (VPC) of the Container Service cluster appears.
VSwitch	Select one or more vSwitches for the Container Service cluster. You can select up to three vSwitches that are deployed in different zones.
Billing Method	Only pay-as-you-go nodes are supported.
Existing Worker Nodes	The number of existing workers in the Container Service cluster.
Nodes to Add	Set the number of worker nodes to add.
Worker Nodes After Scaling	The number of workers after the scaling.
Instance Type	You can select one or more instance types. For more information, see the "*Instance types*" topic of *ECS User Guide*.
Selected Types	The selected instance types.
System Disk	Standard SSDs, enhanced SSDs (ESSDs), and ultra disks are supported.

Parameter	Description
Mount Data Disk	Standard SSDs, ESSDs, and ultra disks are supported.
Operating System	The operating system of the Container Service cluster.
Password	○ **Password**: Enter the password that is used to log on to the nodes. ○ **Confirm Password**: Enter the password again.
RDS Whitelist	Set the RDS whitelist. Add the IP addresses of the scaled nodes to the RDS whitelist.
Label	Add labels to the nodes.
Custom Image	You can specify a custom Elastic Compute Service (ECS) image. This allows you to deploy nodes of the Container Service cluster based on this image.
Taint	Add taints to worker nodes in the Container Service cluster.
CPU Policy	Set the CPU policy. ○ none: the default CPU policy. This policy enables the default CPU affinity scheme. ○ static: This policy allows pods with specific resource characteristics on the node to be configured with enhanced CPU affinity and exclusivity.
User Data	Customize the startup behaviors of Elastic Compute Service (ECS) instances and import data to the ECS instances. The user data can be used to perform the following operations: ○ Run user data scripts during instance startup. ○ Pass user data as common data into an ECS instance for future reference.

5. Click **Submit**.

What's next

After the Container Service cluster is expanded, go to the details page of the Container Service cluster. In the left-side navigation pane, choose **Clusters > Node Pools**. You can find that the number of worker nodes is increased from 3 to 5.

3.1.4.2.6. Renew a certificate

This topic describes how to renew a Kubernetes certificate in the console.

Prerequisites

You have created a Kubernetes cluster and its certificate is about to expire.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters**. The **Clusters** page appears.
3. Select the target cluster and click **Update Certificate**. The **Update Certificate** page appears.

> **Note** The **Update Certificate** button will be displayed two months before your cluster certificate expires.

4. Click **Update** and the **Confirm** page appears.
5. Click **Confirm**.

Result

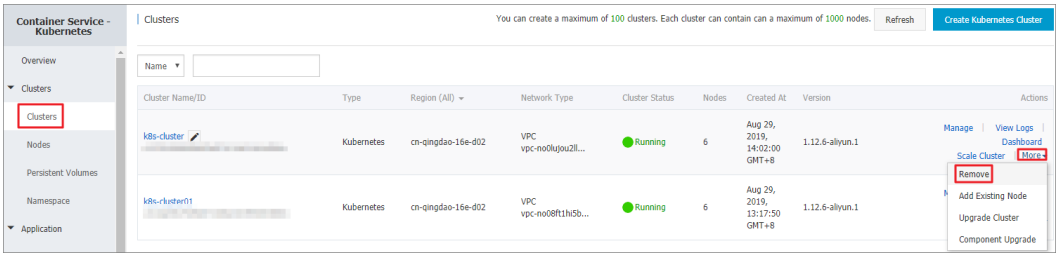
- On the **Update Certificate** page, the following message appears: **The certificate has been updated**.
- On the **Clusters** page, the **Update Certificate** button has disappeared.

3.1.4.2.7. Delete a cluster

You can delete clusters in the Container Service console.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters**. The Clusters page appears.
3. Find the target cluster and choose **More > Delete** in the Actions column.



What's next

Resource Orchestration Service (ROS) has no permission to delete resources that were manually added under ROS-created resources. For example, if you manually add a VSwitch under a ROS-created VPC instance, ROS cannot delete the VPC instance and therefore the cluster cannot be deleted.

Container Service allows you to force delete clusters. If your first attempt to delete a cluster fails, you can forcibly delete the cluster and ROS stack. However, you still need to manually release the resources that were manually added in the first place.

An error message appears when an attempt to delete a cluster fails.

Select the cluster that you failed to delete and choose **More > Delete** in the Actions column. In the dialog box that appears, you can view the resources that were manually added. Select the **Force Delete** check box and click **OK** to delete the cluster and ROS resource stack.

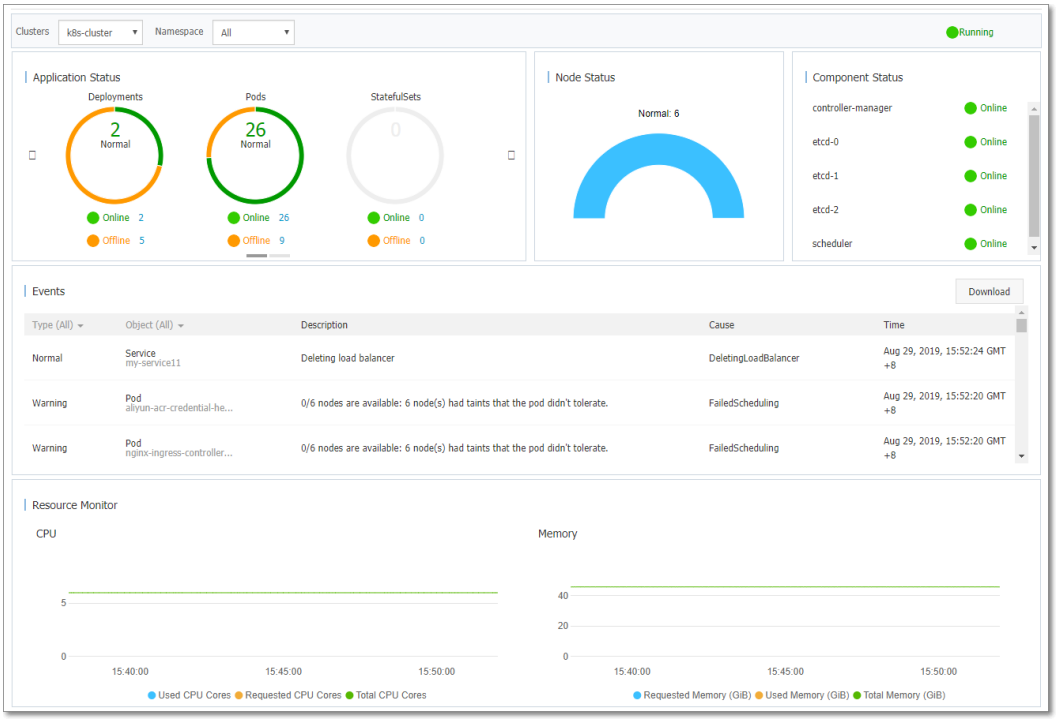
3.1.4.2.8. View cluster overview

The Container Service console provides a cluster overview page. This page displays the information such as application status, component status, and resource monitoring status. This allows you to check the health status of your cluster at your convenience.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, click **Overview**. The Overview page appears.
3. Select the target cluster and namespace. You can view the application status, component status, and resource monitoring charts.
   - **Application Status**: displays the statuses of the deployments, pods, and replica sets that are running in the cluster. Green sections indicate a normal state and yellow sections indicate an exception state.

- **Node Status**: displays the statuses of the nodes in the cluster.
- **Component Status**: Components are deployed in the kube-system namespace. Core components are used, such as the scheduler, controller-manager, and etcd.
- **Events**: displays events such as warnings and errors. If no events are displayed, the cluster is running in the normal state.
- **Monitoring**: displays CPU and memory monitoring charts. CPU usage is measured in cores or millicores and accurate to three decimal places. A millicore is one thousandth of a core. Memory usage is measured in GiB and accurate to three decimal places. For more information, see Meaning of CPU and Meaning of memory.



3.1.4.3. Nodes

3.1.4.3.1. Add an existing node

Container Service allows you to add an existing Elastic Compute Service (ECS) instance to a Kubernetes cluster. You can add only worker nodes to clusters.

Prerequisites

- A Kubernetes cluster is created. For more information, see Log on to the Container Service console.
- An ECS instance is created. Make sure that the region, zone, organization, project, security group, virtual private cloud (VPC), and operating system settings of the ECS instance are the same as those of the cluster.

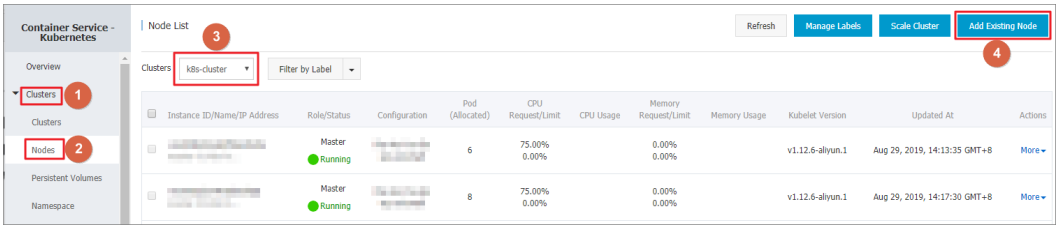
Context

- By default, a cluster can contain up to 50 nodes. To add more nodes to a cluster, submit a ticket.
- The ECS instance must be in the same region and VPC as the cluster.
- The ECS instance must belong to the same Apsara Stack tenant account as the cluster.
- The ECS instance must be running the CentOS operating system.

Procedure

1. Log on to the Container Service console.

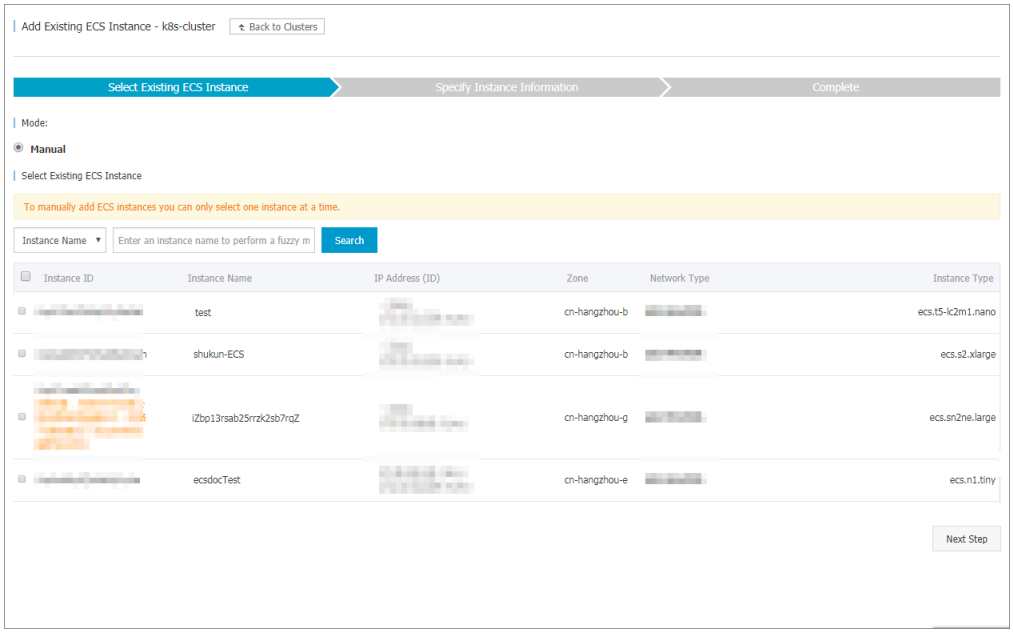
2. In the left-side navigation pane, choose **Clusters > Nodes** to go to the Nodes page.
3. Select the cluster to which you want to add a node and click **Add Existing Node** in the upper-right corner.



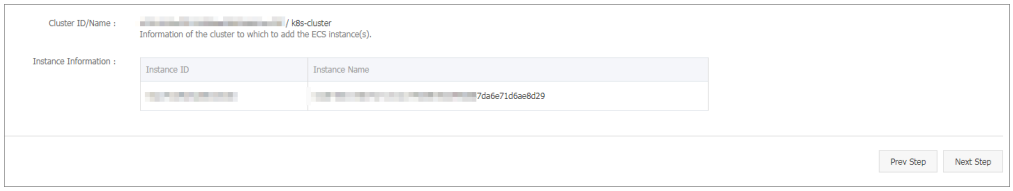
4. On the page that appears, you can manually add existing ECS instances to the cluster.

   To manually add an ECS instance, you must obtain the installation command and log on to the ECS instance to run the command. You can add only one ECS instance at a time.

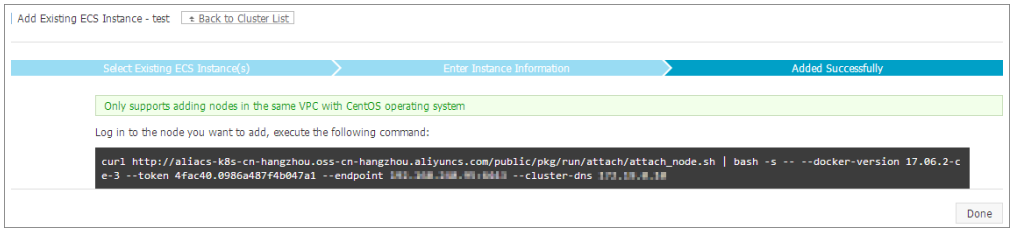
   i. Select the ECS instance that you want to add and click **Next Step**. You can add only one ECS instance at a time.



   ii. Confirm the instance information and click **Next Step**.



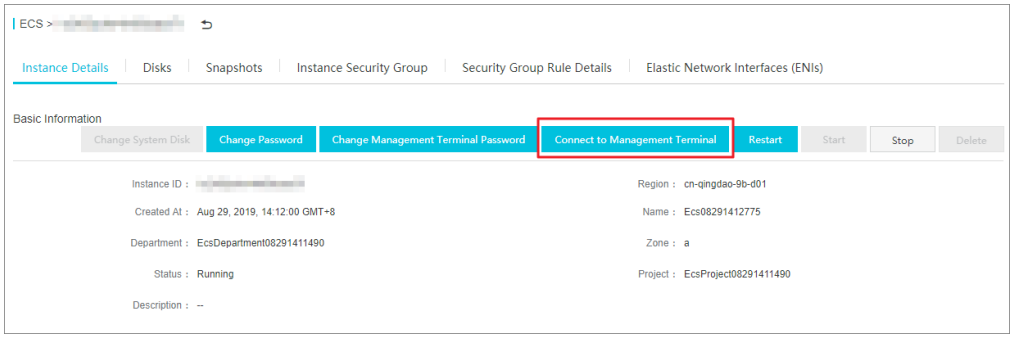
   iii. Copy the command.

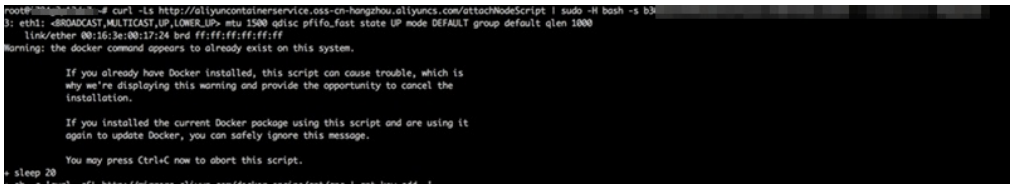


   iv. Click **Complete**.

v. Go to the Apsara Uni-manager Management Console. In the top navigation bar, choose **Products > Elastic Compute Service**. On the **Instances** page, select the organization and region of the cluster, and then find the ECS instance that you want to add to the cluster.

vi. Click the instance name to go to the Instance Details tab. Click **Connect to VNC**. In the dialog box that appears, enter the VNC password and then click **OK**. After you log on to the instance, paste the copied command and click **OK** to run the script.





vii. After you run the script, the ECS instance is added to the cluster. You can go to the Clusters page and click the cluster ID to view nodes in the cluster. Check whether the ECS instance has been added to the cluster.

3.1.4.3.2. View nodes

You can view the nodes in a cluster through commands, the console, or the Kubernetes Dashboard.

Through commands

> **Note** To view the nodes in a cluster through commands, you need to Connect to a Kubernetes cluster through kubectl.

Use kubectl to connect to a cluster and run the following command to view the nodes in the cluster.

```
kubectl get nodes
```

A sample output is as follows:

```
$ kubectl get nodes
    NAME                       STATUS    AGE       VERSION
    iz2ze2n6ep53tch701yh9zz    Ready     19m       v1.6.1-2+ed9e3d33a07093
    iz2zeafr762wibijx39e5az    Ready     7m        v1.6.1-2+ed9e3d33a07093
    iz2zeafr762wibijx39e5bz    Ready     7m        v1.6.1-2+ed9e3d33a07093
    iz2zef4dnn9nos8elyr32kz    Ready     14m       v1.6.1-2+ed9e3d33a07093
    iz2zeitvvo8enoreufstkmz    Ready     11m       v1.6.1-2+ed9e3d33a07093
```

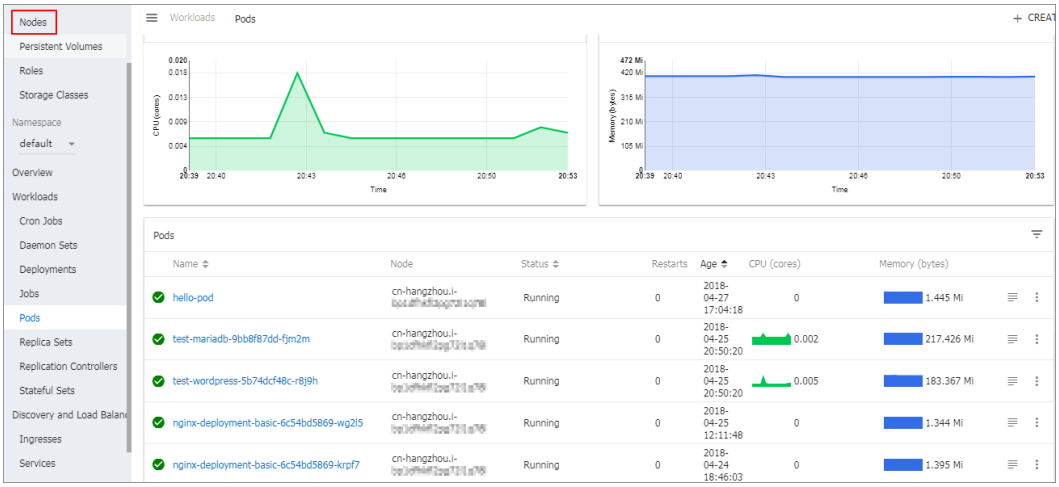
Through the Container Service console

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Nodes** to go to the Nodes page.

3. Select the target cluster to view the nodes in the cluster.

Through the Kubernetes Dashboard

1. Log on to the Container Service console.
2. In the left-side navigation pane, click **Clusters** to go to the Clusters page.
3. Select the target cluster and click **Dashboard** in the Actions column to go to the Kubernetes Dashboard.
4. In the left-side navigation pane, click **Nodes**. On the page that appears, you can view all nodes in the cluster.



3.1.4.3.3. Manage node labels

You can manage node labels through the console. You can add a label to multiple nodes at the same time, filter nodes by label, and remove labels.

For more information about how to use labels to schedule nodes, see Set node scheduling.

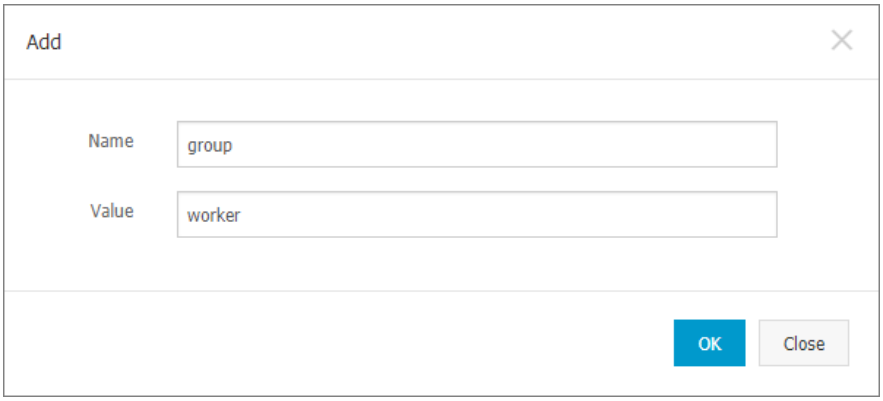
Prerequisites

You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.

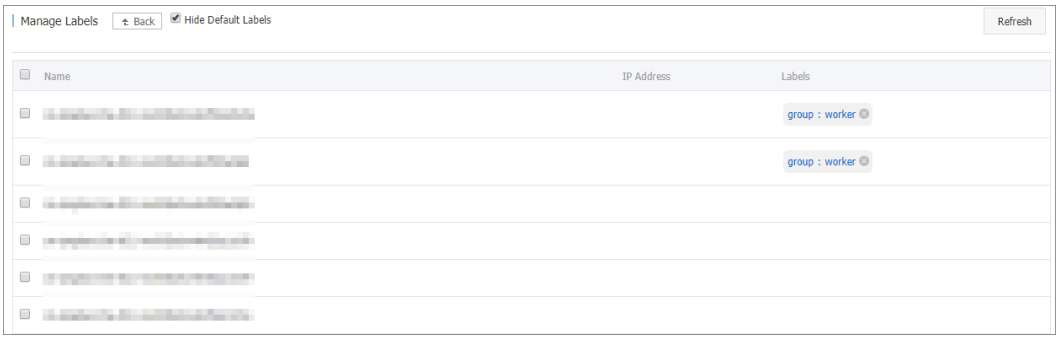
Add a label to multiple nodes

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Nodes** to go to the Nodes page.
3. Select the target cluster and click **Manage Labels** in the upper-right corner.
4. Select multiple nodes and then click **Add Label**.

5. In the dialog box that appears, enter the label name and value, and then click **OK**.

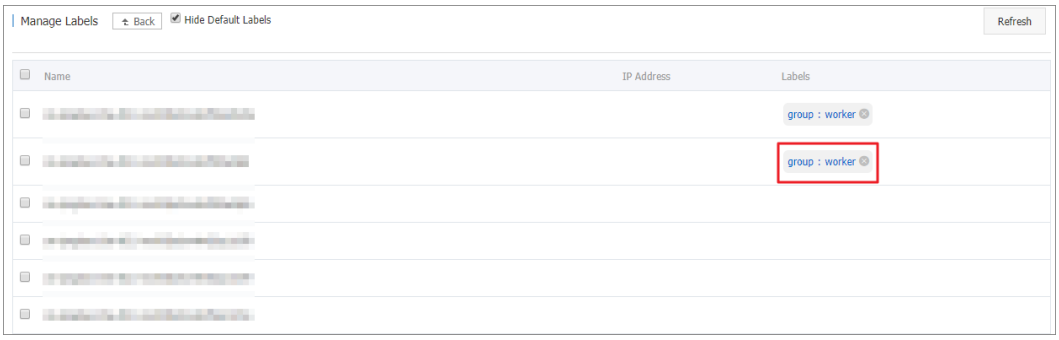


On the Manage Labels page, the selected nodes now have the same label.



Remove a label

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Nodes** to go to the Nodes page.
3. Select the target cluster and click **Manage Labels** in the upper-right corner.
4. Select a node and click the cross sign at the end of a label, for example, `group:worker`.

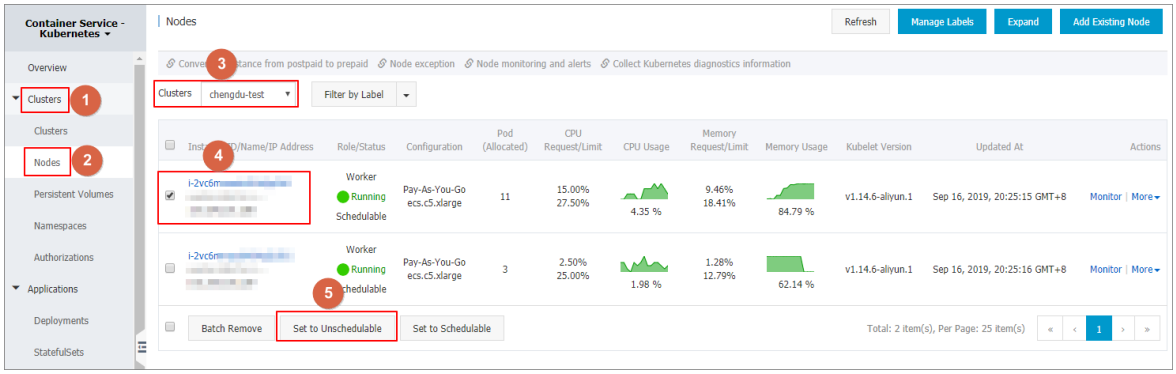


Click **Confirm** to remove the label.

3.1.4.3.4. Set node schedulability

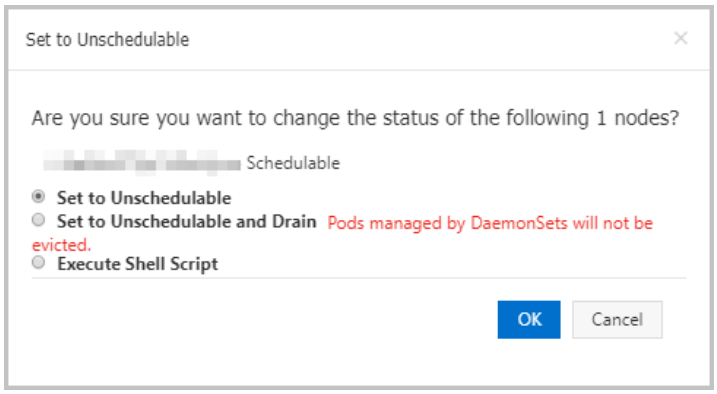
Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Nodes** to go to the Nodes page.
3. Select the target cluster and select one or more nodes. Click **Set to Unschedulable**.



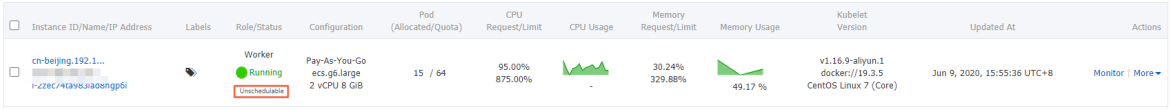
4. In the dialog box that appears, you can set the specified nodes to either of the following statues:
   - **Set to Unschedulable**: Pods will not be scheduled to this node when you deploy new applications.
   - **Set to Unschedulable and Drain**: Pods will not be scheduled to this node when you deploy new applications. Pods on this node will be evicted, except for the pods that are managed by DaemonSets.

     In this example, **Set to Unschedulable** is selected.



5. Click **OK**.

   The node status is now changed to Unschedulable.



What's next

Pods will not be scheduled to the node when you deploy new applications.

3.1.4.3.5. Remove a node

To restart or release an ECS node in a cluster, you must remove the node from the cluster first. This topic describes how to remove a node.

Prerequisites

- You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.
- You can use kubectl to connect to the Kubernetes cluster. For more information, see Connect to a Kubernetes cluster through kubectl.

Context

- When you remove a node, pods running on the node will be migrated to other nodes, which may cause service interruptions. We recommend that you remove nodes during off-peak hours.
- Unexpected errors may occur when you remove a node. We recommend that you back up your data in advance.

- The node that you choose to remove will be set to the unschedulable state.
- You can only remove worker nodes.

Procedure

1. Run the following command to migrate the pods on the target node to other nodes.

   > **Note** Make sure that the other nodes have sufficient resources.

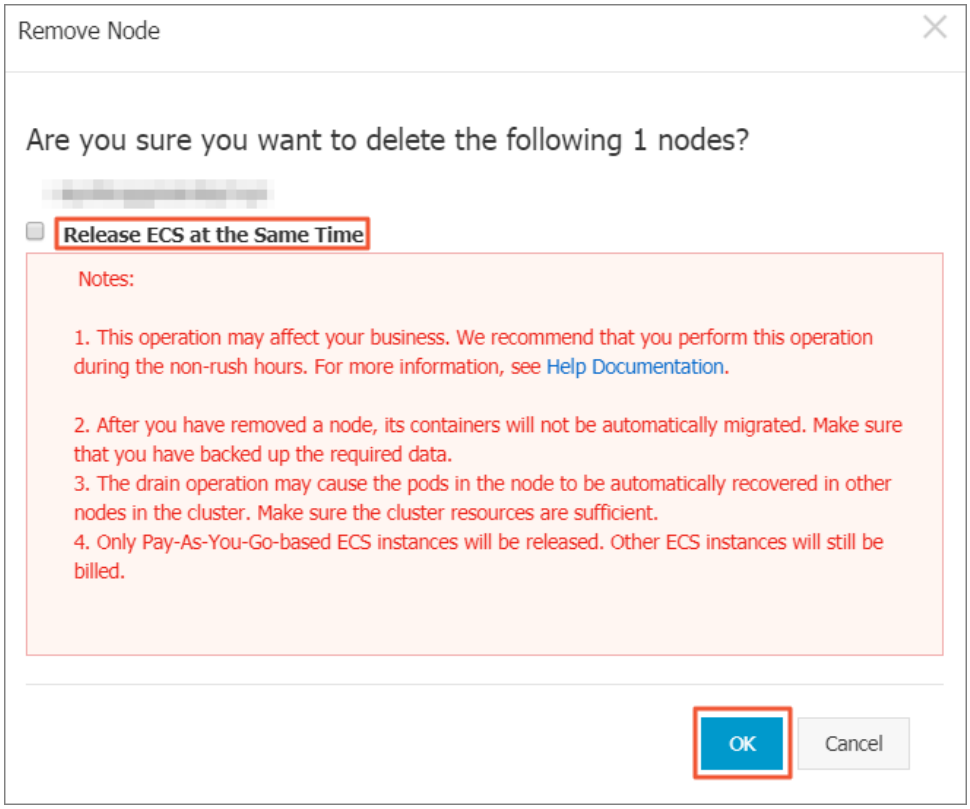
   ```
   kubectl drain node-name
   ```

   > **Note** *node-name* must be in the format of *your-region-name.node-id*.
   > - *your-region-name* represents the region where your cluster is deployed.
   > - *node-id* represents the ID of the ECS instance where the target node is deployed. For example, *cn-hanghzou.i-xxx*.

2. In the left-side navigation pane, choose **Clusters > Nodes** to go to the **Nodes** page.
3. Select the target cluster. Find the target node and choose **More > Remove** in the Actions column. The **Remove Node** page appears.

   > **Note** To remove multiple nodes at the same time, select nodes on the **Nodes** page and click **Batch Remove**.

4. (Optional)To release the ECS instance where the target node is deployed, select the **Release ECS Instance** check box.



> **Note**
> - This option only releases pay-as-you-go ECS instances.
> - Subscription ECS instances will be automatically released after the subscription expires.
> - If you do not select the **Release ECS Instance** check box, you will continue to be billed for the ECS instance where the target node is deployed.

5. Click **OK** to remove the node.

3.1.4.3.6. View node resource usage

You can view the resource usage of the nodes in a cluster through the console.

Prerequisites

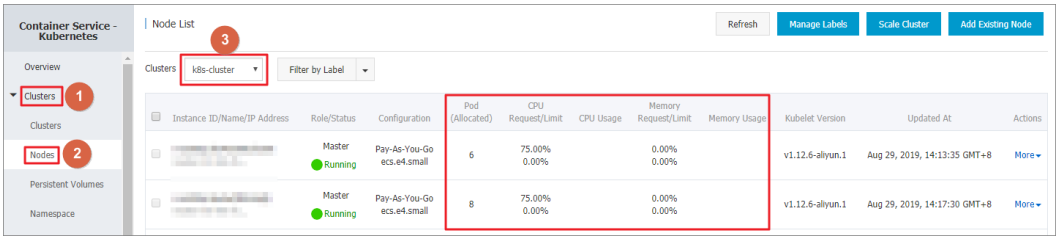
You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Nodes** to go to the Nodes page.

   You can view the request and usage rate of CPU and memory on each node.
   - CPU request rate = sum (The amount of CPU requested by all pods on the node) / Total CPU of the node
   - CPU usage rate = sum (The amount of CPU used by all pods on the node) / Total CPU of the node
   - Memory request rate = (The amount of memory requested by all pods on the node) / Total memory of the node
   - Memory usage rate = sum (The amount of memory used by all pods on the node) / Total memory of the node

> **Note**
> - You can adjust the workload of a node based on resource usage. For more information, see Set node scheduling.
> - When the request or usage rate of a node reaches 100%, pods will not be scheduled to the node.



3.1.4.3.7. Upgrade the NVIDIA driver on a GPU node

This topic describes how to upgrade the NVIDIA driver on a GPU node when workloads are deployed on the node and when no workload is deployed on the node.

Upgrade the NVIDIA driver on a GPU node where workloads are deployed

1. Connect to a Kubernetes cluster through kubectl.
2. Run the following command to set the target node to unschedulable.

```
kubectl cordon node-name
```

> **Note**
>
> - Currently, you can only upgrade the NVIDIA driver on worker nodes.
> - *node-name* must be in the format of *your-region-name.node-id*.
>   - *your-region-name* represents the region where your cluster is deployed.
>   - *node-id* represents the ID of the ECS instance where the target node is deployed.
>
> You can run the following command to query *node-name*.
>
> ```
> kubectl get node
> ```

```
[root@gpu-test ~]# kubectl cordon cn-hangzhou.i-
node/cn-hangzhou.i-                    already cordoned
```

3. Run the following command to migrate pods from the target node to other nodes:

```
kubectl drain node-name --grace-period=120 --ignore-daemonsets=true
```

```
[root@gpu-test ~]# kubectl drain cn-hangzhou.i-                    --grace-period=120 --ignore-daemonsets=true
node/cn-hangzhou.i-                    cordoned
WARNING: Ignoring DaemonSet-managed pods: flexvolume-      , kube-flannel-ds-      , kube-proxy-worker-      , logtail-ds-
pod/domain-nginx-                    evicted
pod/old-nginx-                    evicted
pod/new-nginx-                    evicted
pod/old-nginx-                    evicted
```

4. Run the following command to log on to the target node:

```
ssh root@xxx.xxx.x.xx
```

5. Run the following command to check the current NVIDIA driver version:

```
nvidia-smi
```

```
[root@                    ~]# nvidia-smi
Fri Jan 18 16:44:52 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 384.111                Driver Version: 384.111                   |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P4            On   | 00000000:00:08.0 Off |                    0 |
| N/A   24C    P8     6W /  75W |      0MiB /  7606MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

6. Run the following commands to uninstall the existing driver:

> **Note**
> - If your driver version is *384.111*, perform the following steps.
> - If your driver version is not *384.111*, download the corresponding driver from the official NVIDIA website first.

```
cd /tmp
```

```
curl -O https://cn.download.nvidia.cn/tesla/384.111/NVIDIA-Linux-x86_64-384.111.run
```

```
chmod u+x NVIDIA-Linux-x86_64-384.111.run
```

```
. /NVIDIA-Linux-x86_64-384.111.run --uninstall -a -s -q
```

7. Run the following command to restart the target node:

```
reboot
```

8. Download the driver that you want to use from the official NVIDIA website. In this example, version *410.79* is used.
9. Run the following command to install the downloaded driver under the directory where it was saved:

```
sh . /NVIDIA-Linux-x86_64-410.79.run -a -s -q
```

10. Run the following commands to configure the driver:

```
nvidia-smi -pm 1 || true
```

```
nvidia-smi -acp 0 || true
```

11. Run the following commands to update device-plugin:

```
mv /etc/kubernetes/manifests/nvidia-device-plugin.yml /
```

```
mv /nvidia-device-plugin.yml /etc/kubernetes/manifests/
```

12. Log on to a master node and run the following command to set the target node to schedulable:

```
kubectl uncordon node-name
```

Result

Run the following command on a master node to check the NVIDIA driver version on the target node. The driver version is now *410.79*.

> **Note** Replace *node-name* with the target node name.

```
kubectl exec -n kube-system -t nvidia-device-plugin-node-name nvidia-smi
```

```
[root@gpu-test ~]#  kubectl exec -n kube-system -t nvidia-device-plugin-cn-[redacted] nvidia-smi
Mon Jan 21 03:14:48 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 410.79       Driver Version: 410.79       CUDA Version: N/A      |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P4            On   | 00000000:00:08.0 Off |                    0 |
| N/A   21C    P8     6W /  75W |      0MiB /  7611MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Upgrade the NVIDIA driver on a GPU node where no workload is deployed

1. Run the following command to log on to the target node:

   ```
   ssh root@xxx.xxx.x.xx
   ```

2. Run the following command to check the current NVIDIA driver version:

   ```
   nvidia-smi
   ```

   ```
   [root@[redacted] ~]# nvidia-smi
   Fri Jan 18 16:44:52 2019
   +-----------------------------------------------------------------------------+
   | NVIDIA-SMI 384.111                Driver Version: 384.111                   |
   |-------------------------------+----------------------+----------------------+
   | GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
   | Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
   |===============================+======================+======================|
   |   0  Tesla P4            On   | 00000000:00:08.0 Off |                    0 |
   | N/A   24C    P8     6W /  75W |      0MiB /  7606MiB |      0%      Default |
   +-------------------------------+----------------------+----------------------+

   +-----------------------------------------------------------------------------+
   | Processes:                                                       GPU Memory |
   |  GPU       PID   Type   Process name                             Usage      |
   |=============================================================================|
   |  No running processes found                                                 |
   +-----------------------------------------------------------------------------+
   ```

3. Run the following commands to uninstall the existing driver:

   > **Note**
   > - If your driver version is *384.111*, perform the following steps.
   > - If your driver version is not *384.111*, download the corresponding driver from the official NVIDIA website first.

   ```
   cd /tmp
   ```

   ```
   curl -O https://cn.download.nvidia.cn/tesla/384.111/NVIDIA-Linux-x86_64-384.111.run
   ```

   ```
   chmod u+x NVIDIA-Linux-x86_64-384.111.run
   ```

   ```
   . /NVIDIA-Linux-x86_64-384.111.run --uninstall -a -s -q
   ```

4. Run the following command to restart the target node.

```
reboot
```

5. Download the driver that you want to use from the official NVIDIA website. In this example, version *410.79* is used.
6. Run the following command to install the downloaded driver under the directory where it was saved:

```
sh . /NVIDIA-Linux-x86_64-410.79.run -a -s -q
```

7. Run the following commands to configure the driver:

```
nvidia-smi -pm 1 || true
```

```
nvidia-smi -acp 0 || true
```

Result

Run the following command on a master node to check the NVIDIA driver version on the target node. The driver version is now *410.79*.

> **Note** Replace *node-name* with the target node name.

```
kubectl exec -n kube-system -t nvidia-device-plugin-node-name nvidia-smi
```

```
[root@gpu-test ~]#  kubectl exec -n kube-system -t nvidia-device-plugin-cn-[redacted] nvid
ia-smi
Mon Jan 21 03:14:48 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 410.79       Driver Version: 410.79       CUDA Version: N/A      |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P4            On   | 00000000:00:08.0 Off |                    0 |
| N/A   21C    P8     6W /  75W |      0MiB /  7611MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

3.1.4.3.8. Create a Kubernetes cluster for GPU computing

This topic describes how to create a Kubernetes cluster for GPU computing.

Prerequisites

The Container Service for Kubernetes, Resource Orchestration Service (ROS), and Resource Access Management (RAM) services are activated.

> **Note** Container Service for Kubernetes uses ROS to deploy applications in Kubernetes clusters. Before you create a Kubernetes cluster, you must activate ROS.

Context

Starting from version 1.8, Kubernetes adds support for the following hardware acceleration devices by using device plug-ins: NVIDIA GPUs, InfiniBand devices, and field-programmable gate arrays (FPGAs). GPU solutions developed by the community will be phased out in version 1.10, and removed from the master code in version 1.11. Container Service for Kubernetes enables you to use a GPU-accelerated Kubernetes cluster to run compute-intensive tasks such as machine learning and image processing. You can deploy applications and achieve auto

scaling without the need to install NVIDIA drivers or Compute Unified Device Architecture (CUDA) in advance.

You must complete the following operations in the Container Service console to create a Kubernetes cluster:

- Create Elastic Compute Service (ECS) instances, configure a public key to enable SSH logon from master nodes to other nodes, and configure the Kubernetes cluster by using cloud-init.
- Create a security group that allows access to the Virtual Private Cloud (VPC) network over Internet Control Message Protocol (ICMP).
- Create a VPC network and a VSwitch and create SNAT rules for the VSwitch if you do not specify an existing VPC network.
- Create VPC routing rules.
- Create a NAT gateway and an elastic IP address.
- Create a RAM user and grant it permissions to query, create, and delete ECS instances and permissions to add and delete cloud disks. The RAM user is also granted all permissions on Server Load Balancer (SLB), Cloud Monitor, VPC, Log Service, and Network Attached Storage (NAS). The Kubernetes cluster dynamically creates SLB instances, cloud disks, and VPC routing rules based on your settings.
- Create an internal SLB instance and open port 6443.
- Create a public SLB instance and open ports 6443, 8443, and 22. If you choose to enable SSH logon when you create the cluster, port 22 is enabled. Otherwise, port 22 is not enabled.

Limits

- Kubernetes clusters support only VPC networks.
- By default, each account has specific quotas on the amount of cloud resources that can be created. You cannot create clusters if the quota limit is exceeded. Make sure that you have sufficient quotas before you create a cluster. To request a quota increase, submit a ticket.
  - You can create up to five clusters across all regions for an account. A cluster can contain up to 40 nodes. To create more clusters or nodes, submit a ticket.

    > **Note** In a Kubernetes cluster, you can create up to 48 route entries per VPC. This means that a cluster can contain up to 48 nodes. To increase the number of nodes, submit a ticket to increase the number of route entries first.

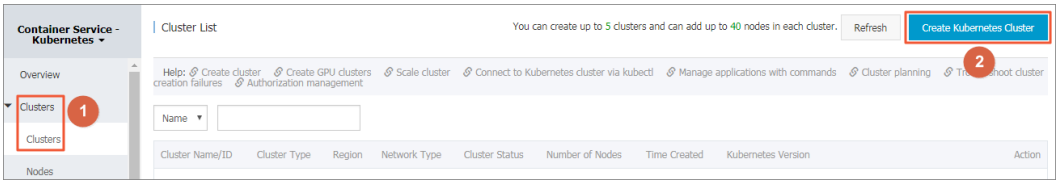
  - You can create up to 100 security groups for each account.
  - You can create up to 60 pay-as-you-go SLB instances for each account.
  - You can create up to 20 elastic IP addresses for each account.
- ECS instances have the following limit:

  Only CentOS is supported.

Create a GN5 Kubernetes cluster

GN5 Kubernetes clusters support only Kubernetes 1.12.6-aliyun.1. Kubernetes 1.11.5 is not supported.

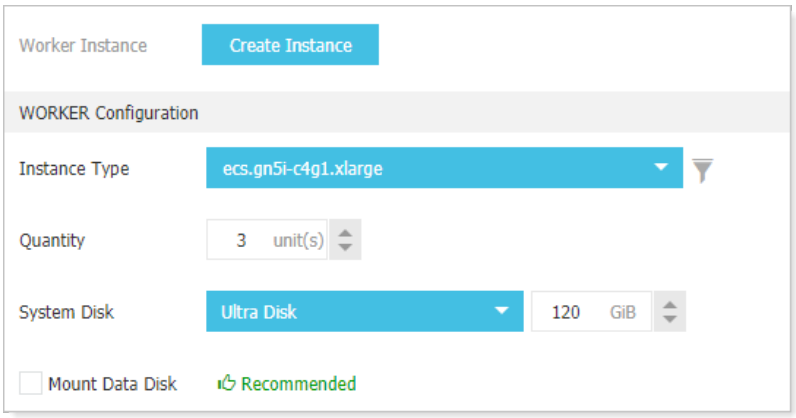
1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters**. The **Clusters** page appears.
3. In the upper-right corner of the page, click **Create Kubernetes Cluster**. The **Create Kubernetes Cluster** page appears.



You are redirected to the **Dedicated Kubernetes** tab by default.

> **Note** To create a cluster for GPU computing, select ECS instance types with GPU capabilities to create worker nodes. For more information about other parameters, see Cluster parameters.

4. Configure worker nodes. In this example, worker nodes are used to run GPU tasks and the gn5i-c4gi1 instance type is seleted.
   i. If you choose to create worker instances, you must set Instance Type and Quantity. Three worker nodes with GPU capabilities are created in this example.



> **Note** We recommend that you use SSD disks.

   ii. If you choose to add existing instances, you must create GPU-accelerated instances in the target region in advance.
5. Set the other parameters and click **Create Cluster** to start the deployment.

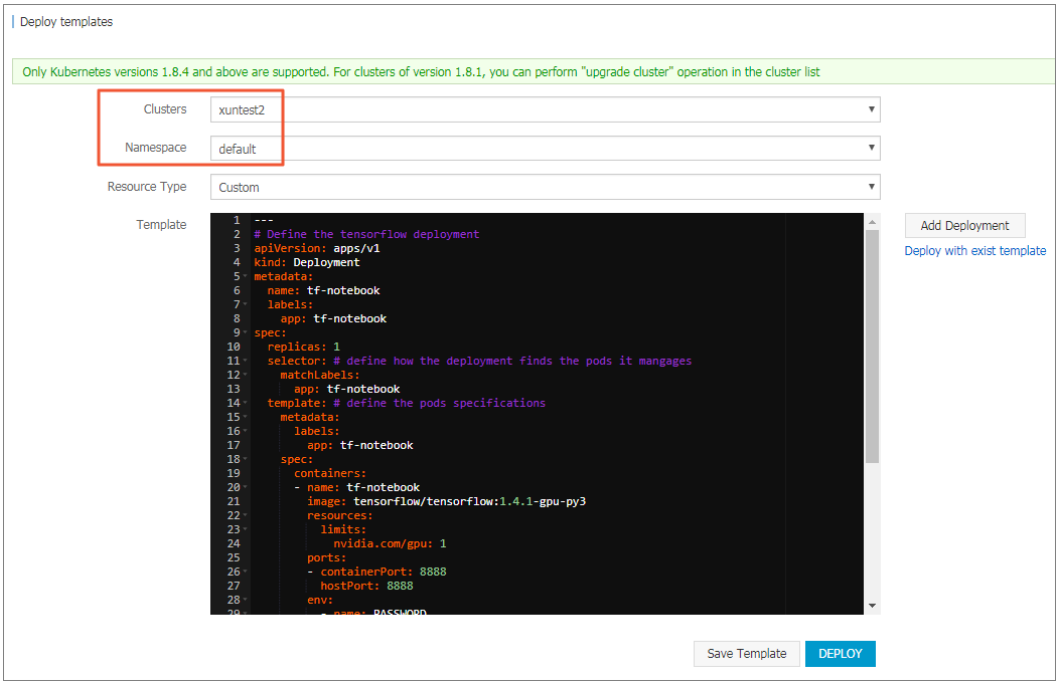
   After the cluster is created, choose **Clusters > Nodes** to go to the Nodes page.

   Select the target cluster. Find one of the created nodes and choose **More > Details** to view the GPU-based devices attached to the node.

Create a GPU experimental environment to run TensorFLow

Jupyter is a standard tool that is used by data scientists to create the experimental environment to run TensorFlow. The following example shows how to deploy a Jupyter application.

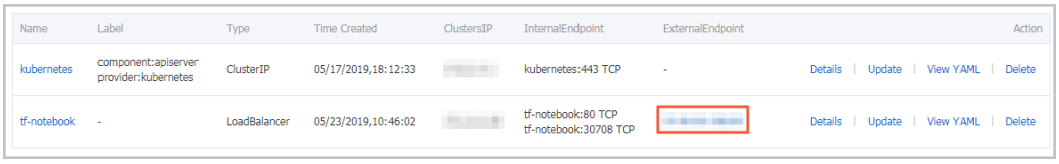
1. In the left-side navigation pane, choose **Applications > Deployments** to go to the **Deployments** page.
2. In the upper-right corner of the page, click **Create from Template**.
3. Select the target cluster and namespace. Select a sample template, or set Sample Template to Custom and customize the template in the Template field. Then, you can click **Create**.



In this example, a Jupyter application template is implemented. The template includes a deployment and a service.

```
---
# Define the tensorflow deployment
apiVersion: apps/v1
kind: Deployment
metadata:
  name: tf-notebook
  labels:
    app: tf-notebook
spec:
  replicas: 1
  selector: # define how the deployment finds the pods it manages
    matchLabels:
      app: tf-notebook
  template: # define the pods specifications
    metadata:
      labels:
        app: tf-notebook
    spec:
      containers:
      - name: tf-notebook
        image: tensorflow/tensorflow:1.4.1-gpu-py3
        resources:
          limits:
            nvidia.com/gpu: 1                    #Specifies the number of NVIDIA GPUs that are
called by the application.
        ports:
        - containerPort: 8888
          hostPort: 8888
        env:
          - name: PASSWORD                       #Specifies the password used to access the Jup
yter instance. You can modify the password as required.
            value: mypassw0rd
# Define the tensorflow service
---
apiVersion: v1
kind: Service
metadata:
  name: tf-notebook
spec:
  ports:
  - port: 80
    targetPort: 8888
    name: jupyter
  selector:
    app: tf-notebook
  type: LoadBalancer                             #Creates an SLB service to ensure that the Jupyter
instance is accessible over the Internet.
```

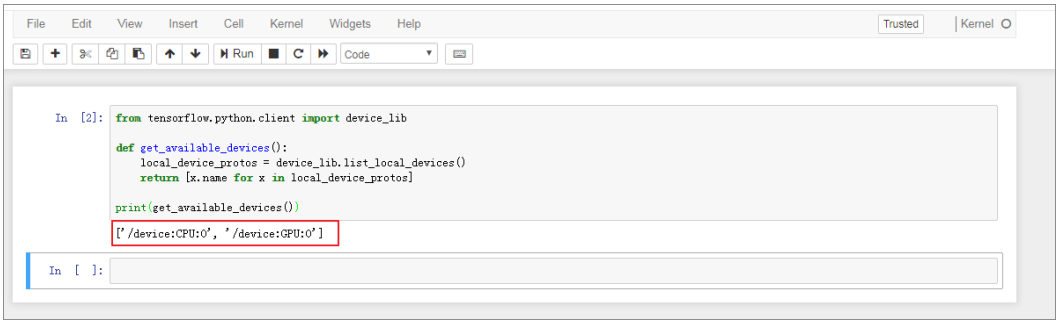
4. In the left-side navigation pane, choose **Ingresses and Load Balancing > Services**. Select the target cluster and namespace. Find the tf-notebook service and check its external endpoint.



5. To connect to the Jupyter instance in a browser, enter `http://EXTERNAL-IP` in the address bar and enter the password specified in the template.

6. You can run the following program to verify that the Jupyter instance has access to GPU-based devices. The program lists all devices that can be used by TensorFlow:

```
from tensorflow.python.client import device_lib
def get_available_devices():
    local_device_protos = device_lib.list_local_devices()
    return [x.name for x in local_device_protos]
print(get_available_devices())
```



3.1.4.3.9. Use labels to schedule pods to GPU nodes

To use Kubernetes clusters for GPU computing, you need to schedule pods to GPU nodes. To make the scheduling more flexible and efficient, you can add labels to GPU nodes.

Context

When Kubernetes deploys nodes with NVIDIA GPUs, the attributes of these GPUs will be discovered and exposed as node labels, which have the following benefits:

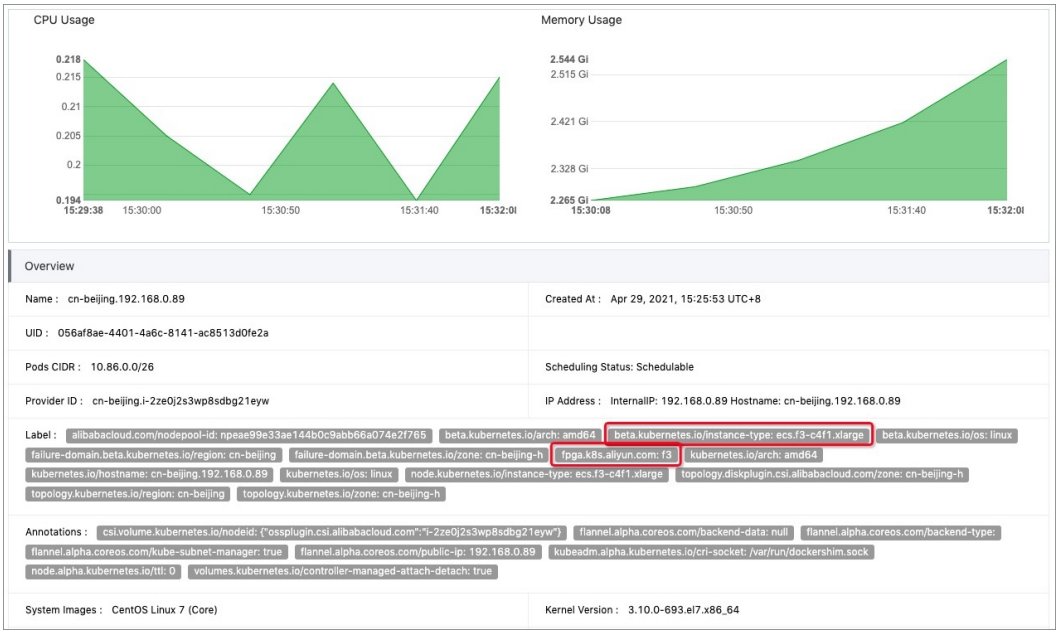
- You can quickly filter GPU nodes by label.
- You can use labels as scheduling conditions when you deploy pods.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Nodes**. The Nodes page appears.

   > **Note** This example selects a cluster with three worker nodes, among which two are equipped with GPUs. Note the node IP addresses.

3. Select a GPU node and choose **More > Details** in the Actions column to go to Kubernetes Dashboard. You can view the labels attached to the node.



You can also log on to a master node and run the following command to view the labels on GPU nodes:

```
# kubectl get nodes
NAME                                STATUS    ROLES     AGE       VERSION
cn-beijing.i-2ze2dy2h9w97v65u****   Ready     master    2d        v1.12.6-aliyun.1
cn-beijing.i-2ze8o1a45qdv5q8a****   Ready     <none>    2d        v1.12.6-aliyun.1            #
Compare these nodes with the nodes displayed in the console to identify GPU nodes.
cn-beijing.i-2ze8o1a45qdv5q8a****   Ready     <none>    2d        v1.12.6-aliyun.1
cn-beijing.i-2ze9xylyn11vop7g****   Ready     master    2d        v1.12.6-aliyun.1
cn-beijing.i-2zed5sw8snjniq6m****   Ready     master    2d        v1.12.6-aliyun.1
cn-beijing.i-2zej9s0zijykp9pw****   Ready     <none>    2d        v1.12.6-aliyun.1
```

Select a GPU node and run the following command to query its labels:

```
# kubectl describe node cn-beijing.i-2ze8o1a45qdv5q8a****
Name:               cn-beijing.i-2ze8o1a45qdv5q8a7luz
Roles:              <none>
Labels:             aliyun.accelerator/nvidia_count=1                        # This field is im
portant.
                    aliyun.accelerator/nvidia_mem=12209MiB
                    aliyun.accelerator/nvidia_name=Tesla-M40
                    beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/instance-type=ecs.gn4-c4g1.xlarge
                    beta.kubernetes.io/os=linux
                    failure-domain.beta.kubernetes.io/region=cn-beijing
                    failure-domain.beta.kubernetes.io/zone=cn-beijing-a
                    kubernetes.io/hostname=cn-beijing.i-2ze8o1a45qdv5q8a****
 ......
```

In this example, the GPU node is attached with the following three labels:

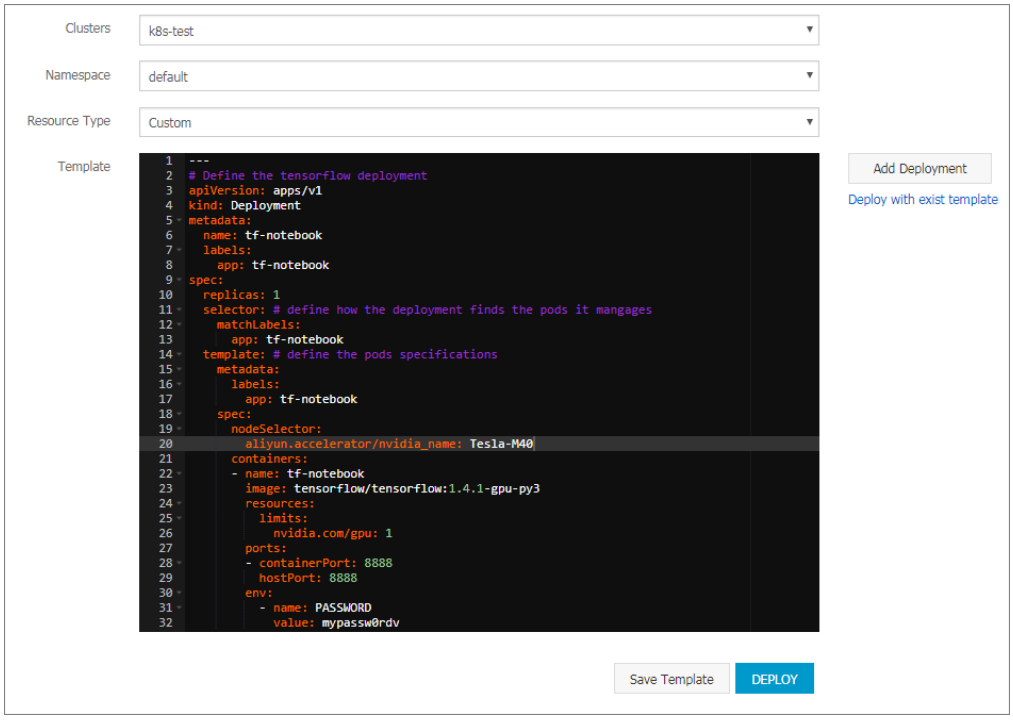
key	value
`aliyun.accelerator/nvidia_count`	The number of GPU cores.
`aliyun.accelerator/nvidia_mem`	The size of the GPU memory in MiB.

key	value
`aliyun.accelerator/nvidia_name`	The name of the NVIDIA graphics card.

GPU nodes of the same type have the same graphics card name. You can use this label to filter nodes.

```
# kubectl get no -l aliyun.accelerator/nvidia_name=Tesla-M40
NAME                                STATUS    ROLES     AGE       VERSION
cn-beijing.i-2ze8o1a45qdv5q8a****   Ready     <none>    2d        v1.12.6-aliyun.1
cn-beijing.i-2ze8o1a45qdv5q8a****   Ready     <none>    2d        v1.12.6-aliyun.1
```

4. Go to the homepage of the Container Service console. In the left-side navigation pane, choose **Applications > Deployments**. On the page that appears, click **Create from Template** in the upper-right corner.
   i. Create a TensorFlow deployment and schedule it to a GPU node.




```
---
# Define the tensorflow deployment
apiVersion: apps/v1
kind: Deployment
metadata:
  name: tf-notebook
  labels:
    app: tf-notebook
spec:
  replicas: 1
  selector: # define how the deployment finds the pods it mangages
    matchLabels:
      app: tf-notebook
  template: # define the pods specifications
    metadata:
      labels:
        app: tf-notebook
    spec:
      nodeSelector:
        aliyun.accelerator/nvidia_name: Tesla-M40
      containers:
      - name: tf-notebook
        image: tensorflow/tensorflow:1.4.1-gpu-py3
        resources:
          limits:
            nvidia.com/gpu: 1
        ports:
        - containerPort: 8888
          hostPort: 8888
        env:
          - name: PASSWORD
            value: mypassw0rdv
```

This example uses the following YAML template:

```
---
# Define the tensorflow deployment
apiVersion: apps/v1
kind: Deployment
metadata:
  name: tf-notebook
  labels:
    app: tf-notebook
spec:
  replicas: 1
  selector: # define how the deployment finds the pods it mangages
    matchLabels:
      app: tf-notebook
  template: # define the pods specifications
    metadata:
      labels:
        app: tf-notebook
    spec:
      nodeSelector:                                              # This field is importan
t.
        aliyun.accelerator/nvidia_name: Tesla-M40
      containers:
      - name: tf-notebook
        image: tensorflow/tensorflow:1.4.1-gpu-py3
        resources:
          limits:
            nvidia.com/gpu: 1                                    # This field is importan
t.
        ports:
        - containerPort: 8888
          hostPort: 8888
        env:
          - name: PASSWORD
            value: mypassw0rdv
```

ii. You can also avoid deploying an application to a GPU node. The following example deploys an NGINX pod and schedules the pod based on node affinity. For more information, see the part about node affinity in Create an application from an image.

This example uses the following YAML template:

```
apiVersion: v1
kind: Pod
metadata:
  name: not-in-gpu-node
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: aliyun.accelerator/nvidia_name
            operator: DoesNotExist
  containers:
  - name: not-in-gpu-node
    image: nginx
```

5. In the left-side navigation pane, choose **Applications > Pods**. On the page that appears, select the target cluster and namespace.

Result

On the Pods page, the two pods from preceding examples have been scheduled to the target nodes. You can use labels to schedule pods to specific GPU nodes with ease.

3.1.4.3.10. Manually upgrade the kernel of a GPU node in a cluster

This topic describes how to manually upgrade the kernel of a GPU node in a cluster.

Context

The current kernel version is earlier than `3.10.0-957.21.3`.

Procedure

1. Connect to a Kubernetes cluster through kubectl.
2. Run the following command to set the target GPU node to unschedulable. This example uses node cn-beijing.i-2ze19qyi8votgjz12345 as the target node.

```
kubectl cordon cn-beijing.i-2ze19qyi8votgjz12345
node/cn-beijing.i-2ze19qyi8votgjz12345 already cordoned
```

3. Run the following command to drain the target GPU node:

```
# kubectl drain cn-beijing.i-2ze19qyi8votgjz12345 --grace-period=120 --ignore-daemonsets=true
node/cn-beijing.i-2ze19qyi8votgjz12345 cordoned
WARNING: Ignoring DaemonSet-managed pods: flexvolume-9scb4, kube-flannel-ds-r2qmh, kube-proxy-wor
ker-l62sf, logtail-ds-f9vbg
pod/nginx-ingress-controller-78d847fb96-5fkkw evicted
```

4. Uninstall the existing nvidia-driver.

> **Note** This step uninstalls the version 384.111 driver. If your driver version is not 384.111, you need to download a driver from the official NVIDIA website and replace `384.111` with your actual version number.

   i. Log on to the target GPU node and run the `nvidia-smi` command to query the driver version.

```
# nvidia-smi -a | grep 'Driver Version'
Driver Version                      : 384.111
```

   ii. Run the following commands to download the driver installation package:

```
cd /tmp/
curl -O https://cn.download.nvidia.cn/tesla/384.111/NVIDIA-Linux-x86_64-384.111.run
```

> **Note** The installation package is required to uninstall the driver.

   iii. Run the following commands to uninstall the existing nvidia-driver:

```
chmod u+x NVIDIA-Linux-x86_64-384.111.run
. /NVIDIA-Linux-x86_64-384.111.run --uninstall -a -s -q
```

5. Run the following commands to upgrade kernel:

```
yum clean all && yum makecache
yum update kernel  -y
```

6. Run the following command to restart the GPU node:

```
reboot
```

7. Log on to the GPU node and run the following command to install the kernel-devel package.

```
yum install -y kernel-devel-$(uname -r)
```

8. Run the following commands to download the required driver and install it on the target node. In this example, version 410.79 is used.

```
cd /tmp/
curl -O https://cn.download.nvidia.cn/tesla/410.79/NVIDIA-Linux-x86_64-410.79.run
chmod u+x NVIDIA-Linux-x86_64-410.79.run
sh . /NVIDIA-Linux-x86_64-410.79.run -a -s -q
# warm up GPU
nvidia-smi -pm 1 || true
nvidia-smi -acp 0 || true
nvidia-smi --auto-boost-default=0 || true
nvidia-smi --auto-boost-permission=0 || true
nvidia-modprobe -u -c=0 -m || true
```

9. Check the */etc/rc.d/rc.local* file and check whether the following configurations are included. If not, add the following content.

```
nvidia-smi -pm 1 || true
nvidia-smi -acp 0 || true
nvidia-smi --auto-boost-default=0 || true
nvidia-smi --auto-boost-permission=0 || true
nvidia-modprobe -u -c=0 -m || true
```

10. Run the following commands to restart kubelet and Docker.

```
service kubelet stop
service docker restart
service kubelet start
```

11. Run the following command to set the GPU node to schedulable:

```
# kubectl uncordon cn-beijing.i-2ze19qyi8votgjz12345
node/cn-beijing.i-2ze19qyi8votgjz12345 already uncordoned
```

12. Run the following command on the nvidia-device-plugin container to check the driver version:

```
kubectl exec -n kube-system -t nvidia-device-plugin-cn-beijing.i-2ze19qyi8votgjz12345 nvidia-smi
Thu Jan 17 00:33:27 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 410.79       Driver Version: 410.79       CUDA Version: N/A      |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  On   | 00000000:00:09.0 Off |                    0 |
| N/A   27C    P0    28W / 250W |      0MiB / 16280MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

3.1.4.3.11. Node pools

3.1.4.3.11.1. Create a node pool

You can add nodes to a node pool. This allows you to manage multiple nodes as a group. For example, you can manage the labels and taints that are added to the nodes in the node pool. This topic describes how to create a node pool in the Container Service console.

Prerequisites

- A Container Service cluster is created. For more information, see Create a Kubernetes cluster.
- The Container Service cluster must run Kubernetes V1.9 or later.

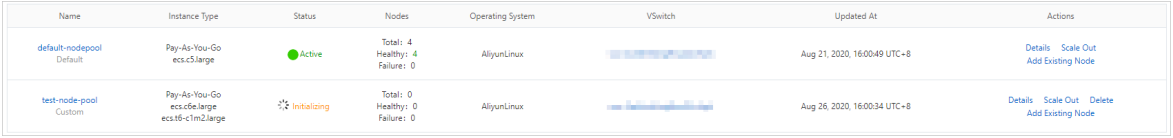
> Notice
> - By default, you can deploy up to 100 nodes in each Container Service cluster. To increase the quota, submit a ticket.
> - Before you add an existing Elastic Compute Service (ECS) instance that is deployed in a virtual private cloud (VPC), make sure that an elastic IP address (EIP) is assigned to the ECS instance. Otherwise, make sure that a NAT gateway is created in the same VPC for the ECS instance. In addition, make sure that the node to be added to the node pool has access to the Internet. Otherwise, you may fail to add the ECS instance.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Node Pools**. On the **Node Pools** page, select the Container Service cluster that you want to manage from the Cluster drop-down list.
3. On the **Node Pools** page, click **Create Node Pool**.
4. In the **Create Node Pool** dialog box, configure the node pool.

   For more information, see Create a Kubernetes cluster. The following list describes some of the parameters:

   - Node Pool Name: Enter a name for the node pool.
   - Quantity: Specify the number of nodes that you want to add to the node pool. If you do not want to add nodes to the pool, set this parameter to 0.
   - ECS Label: Add labels to the ECS instances.
   - Node Label: Add labels to nodes in the node pool.
   - Custom Resource Group: Specify the resource group to which the nodes of the node pool belong.
   - Custom Security Group: Set a custom security group.
5. Click **OK**.
   On the **Node Pools** page, check whether the node pool is in the **Initializing** state. This state indicates that the node pool is being created. After the node pool is created, the node pool is in the **Active** state.



What's next

After the node pool is created, find the node pool on the **Node Pools** page and click **Details** in the **Actions** column to check further details of node pool.

3.1.4.3.11.2. Scale out a node pool

You can manage multiple nodes as a group by adding nodes to a node pool. For example, you can centrally manage the labels and taints on nodes in a node pool. This topic describes how to scale out a node pool in the Container Service console.

Prerequisites

- A node pool is created. For more information, see Create a node pool.
- The Container Service cluster must run Kubernetes V1.9 or later.

> Notice
> - By default, you can deploy up to 100 nodes in each Container Service cluster. To increase the quota, submit a ticket.
> - Before you add an existing ECS instance that is deployed in a virtual private cloud (VPC), make sure that an elastic IP address is assigned to the ECS instance, or a NAT gateway is created in the same VPC for the ECS instance. In addition, the node must have access to the Internet. Otherwise, you may fail to add the ECS instance.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Node Pools**. On the **Node Pools** page, select the cluster that you want to manage from the Cluster drop-down list.
3. On the Node Pools page, find the node pool that you want to scale out and click **Scale Out** in the **Actions** column.
4. In the dialog box that appears, set the number of nodes to be added to the node pool.
5. (Optional)In the dialog box that appears, click **Modify Node Pool** to modify the node pool.

   For more information, see Expand a Container Service cluster. Set the following parameters:

   - ECS Label: You can add labels to the specified ECS instances.
   - Node Label: You can add labels to the specified nodes.
   - Taints: You can add one or more taints to the specified nodes.

     > Note If you select **Synchronize Node Labels and Taints**, you can synchronize the specified labels and taints to the existing and added nodes.

   - Cloud Monitor: You can install the Cloud Monitor agent on the nodes and view monitoring information about the nodes in the Cloud Monitor console.
6. Click **OK**.
   On the **Node Pools** page, the **Status** column displays **Scaling** for the node pool. This indicates that the scale-out event is in progress. After the scale-out event is completed, the **Status** column shows **Active** for the node pool.

What's next

Click **Details** in the Actions column for the node pool. On the **Nodes** tab, you can check the nodes that are added to the node pool of the cluster.

3.1.4.3.11.3. Schedule an application pod to a specified node pool

Labels are an important part of Kubernetes. Services, deployments, and pods are associated with each other by labels. You can set pod scheduling policies related to node labels. This allows you to schedule pods to nodes that have specified labels. This topic describes how to schedule an application pod to a specified node pool.

Procedure

1. Add a label to a node pool.

   In Container Service, you can manage a group of cluster nodes in a node pool. For example, you can manage labels and taints of all nodes in a node pool. For more information about how to create a node pool, see Create a node pool.

   i. Log on to the Container Service console

   ii. In the left-side navigation pane, choose **Clusters > Node Pools**. On the **Node Pools** page, select the cluster that you want to manage from the Cluster drop-down list.

   iii. In the upper-right corner of the **Node Pools** page, click **Create Node Pool**.

   iv. On the Create Node Pool page, click **Show Advanced Options** and then click the  icon to add labels to nodes.

   In this example, the label that is added to the node is pod: nginx.

   You can also click **Scale Out** on the right side of a node pool to update or add labels for nodes. If automatic scaling is enabled for a node pool, click **Modify** on the right side of the node pool to update or add labels for nodes.

2. Set a scheduling policy for the application pod.

   In the preceding step, the pod: nginx label is added to the nodes in the node pool. You can use the nodeSelector or nodeAffinity field to schedule an application pod to a specified node pool. The following content describes how to schedule an application pod:

   - Set nodeSelector.

     nodeSelector is a field in the spec section. Add the pod: nginx label to nodeSelector. You can use the following code block:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment-basic
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      nodeSelector:
        pod: nginx      # After you add the label of a node pool, this application pod can run only on nodes in the node pool.
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

- Set nodeAffinity.

  You can also use nodeAffinity to schedule application pods. The following four scheduling policies are available:

  ```
  - requiredDuringSchedulingIgnoredDuringExecution
  ```

  If this policy is used, the pod must be deployed on a node that meets the requirements. If no nodes meet the requirements, the system retries until a node that meets the requirements is found. IgnoreDuringExecution indicates that if the label of the node where the pod is deployed changes and does not meet the requirements, the pod continues to run on the node.

  ```
  - requiredDuringSchedulingRequiredDuringExecution
  ```

  If this policy is used, the pod must be deployed on a node that meets the requirements. If no nodes meet the requirements, the system retries until a node that meets the requirements is found. RequiredDuringExecution indicates that if the label of the node where the pod is deployed changes and does not meet the requirements, the system selects another node that meets the requirements to deploy the pod.

  ```
  - preferredDuringSchedulingIgnoredDuringExecution
  ```

  If this policy is used, the pod is preferentially deployed on a node that meets the requirements. If no nodes meet the requirements, the system ignores these requirements.

  ```
  - preferredDuringSchedulingRequiredDuringExecution
  ```

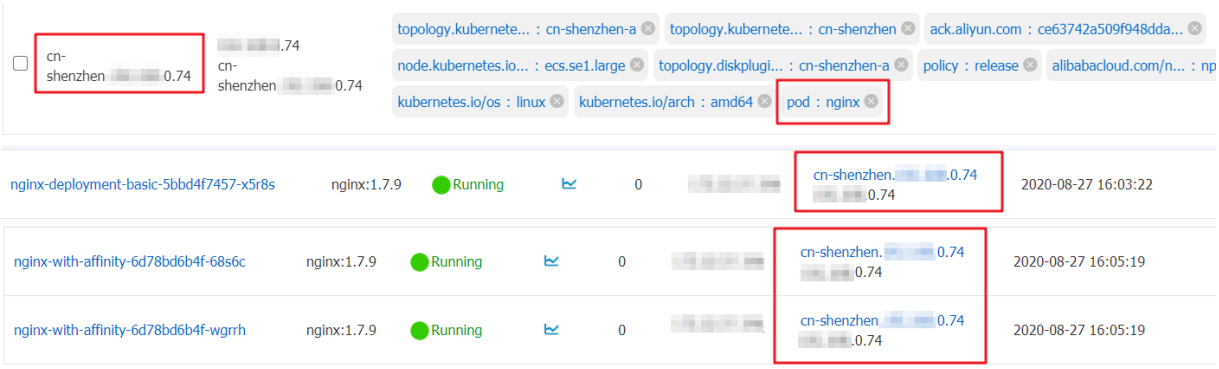
  If this policy is used, the pod is preferentially deployed on a node that meets the requirements. If no nodes meet the requirements, the system ignores these requirements. RequiredDuringExecution indicates that if the label of a node changes and meets the requirements after the pod is deployed on another node, the system deploys the pod on the node that meets the requirements.

In the following example, the requiredDuringSchedulingIgnoredDuringExecution policy is used to ensure that the application runs on a node in the specified node pool.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-with-affinity
  labels:
    app: nginx-with-affinity
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx-with-affinity
  template:
    metadata:
      labels:
        app: nginx-with-affinity
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: pod
                operator: In      # The application runs on a node that has the pod: nginx lab
el.
                values:
                - nginx
      containers:
      - name: nginx-with-affinity
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

Result

All application pods in the preceding examples are scheduled to the xxx.xxx.0.74 node. This node has the pod: nignx label.



3.1.4.4. Storage

3.1.4.4.1. Overview

In the Container Service console, you can create volumes of other Apsara Stack services, enabling you to create stateful applications and use Apsara Stack disks and OSS to implement persistent storage.

Both static and dynamic volumes are supported. The following table shows how static and dynamic volumes are supported.

Apsara Stack storage	Static volume	Dynamic volume
Apsara Stack disk	You can use a static disk volume through either of the following methods: • Use a volume directly • Use a volume through a PV and PVC	Supported

Apsara Stack NAS	You can use a static NAS volume through either of the following methods:
	• Use a volume through the FlexVolume plug-in
	  ◦ Use a volume directly
	  ◦ Use a volume through a PV or PVC
	• Use a volume through the Kubernetes NFS driver
		Supported

Apsara Stack OSS	You can use a static OSS volume through either of the following methods: • Use a volume directly • Use a volume through a PV or PVC	Not supported

3.1.4.4.2. Mount a disk to a cluster

You can mount disks as volumes.

You can mount disks to Kubernetes clusters as volumes.

Disks can be mounted to Kubernetes clusters as the following types of volume:

- Mount a disk to a cluster as a statically provisioned volume

  You can use statically provisioned volumes in the following ways:

  - Mount a disk as a volume
  - Create a persistent volume (PV) and a persistent volume claim (PVC)

- Mount a disk as a dynamically provisioned volume

Usage Notes

- A disk can be mounted to only one pod.
- Before you mount a disk as a volume to a pod, you must create the disk and obtain its disk ID.

  The disk must meet the following capacity requirements:

  - A basic disk must have a minimum capacity of 5 GiB.
  - An ultra disk must have a minimum capacity of 20 GiB.
  - A standard SSD must have a minimum capacity of 20 GiB.

- volumeId: the ID of the disk that you want to mount. The value must be the same as those of volumeName and PV Name.
- The node and the disk to be mounted must be in the same zone.

- Only pay-as-you-go disks can be mounted. If you change the billing method of an Elastic Compute Service (ECS) instance in the cluster from pay-as-you-go to subscription, you cannot change the billing method of its disks to subscription. Otherwise, the disks cannot be mounted to the cluster.

Mount a disk to a cluster as a statically provisioned volume

You can mount disks as volumes or by creating PVs and PVCs.

Prerequisites

A disk is created in the ECS console.

- **Use a disk as a volume**

  Use the following *disk-deploy.yaml* file to create a pod:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-disk-deploy
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx-flexvolume-disk
        image: nginx
        volumeMounts:
          - name: "d-bp1j17ifxfasvts3tf40"
            mountPath: "/data"
      volumes:
        - name: "d-bp1j17ifxfasvts3tf40"
          flexVolume:
            driver: "alicloud/disk"
            fsType: "ext4"
            options:
              volumeId: "d-bp1j17ifxfasvts3tf40"
```

- **Create a PV and a PVC**

  i. **Create a PV of the disk type**

  You can create a PV of the disk type in the console or by using a YAML file.

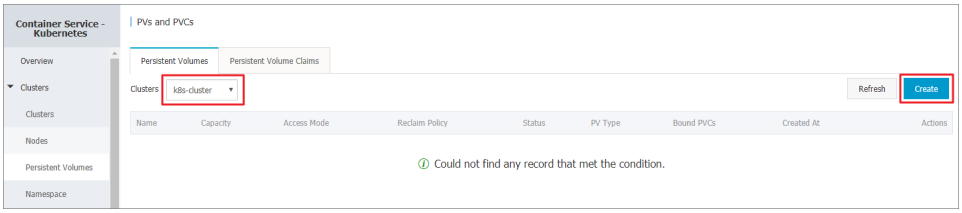
- **Create a PV by using a YAML file**

  Use the following `disk-pv.yaml` file to create a PV:

  > **Note** The PV name must be the same as the disk ID.

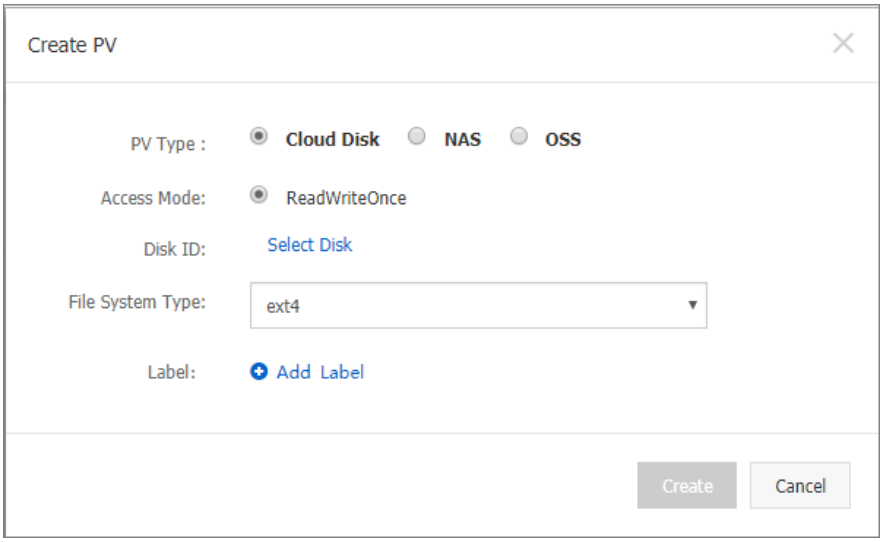
```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: d-bp1j17ifxfasvts3tf40
  labels:
    failure-domain.beta.kubernetes.io/zone: cn-hangzhou-b
    failure-domain.beta.kubernetes.io/region: cn-hangzhou
spec:
  capacity:
    storage: 20Gi
  storageClassName: disk
  accessModes:
    - ReadWriteOnce
  flexVolume:
    driver: "alicloud/disk"
    fsType: "ext4"
    options:
      volumeId: "d-bp1j17ifxfasvts3tf40"
```

- **Create a PV in the console**
  a. Log on to the Container Service console.
  b. In the left-side navigation pane, choose **Clusters > Persistent Volumes** to go to the **Persistent Volumes** page.
  c. On the **Persistent Volumes** tab, select the cluster that you want to manage and click **Create** in the upper-right corner.



  d. Set the parameters in the Create PV dialog box. PV parameters

Parameter	Description
PV Type	In this example, **Cloud Disk** is selected.
Access Mode	By default, ReadWriteOnce is used.
Disk ID	Select a disk that is in the same region and zone as the cluster.
File System Type	Select the file system of the disk. Supported file systems include ext4, ext3, xfs, and vfat. Default value: ext4.
Label	Add labels to the PV.



  e. Click **Create**.

ii. **Create a PVC**

Use the following `disk-pvc.yaml` file to create a PVC:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-disk
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: disk
  resources:
    requests:
      storage: 20Gi
```

iii. **Create a pod**

Use the following *disk-pod.yaml* file to create a pod:

```
apiVersion: v1
kind: Pod
metadata:
  name: "flexvolume-alicloud-example"
spec:
  containers:
    - name: "nginx"
      image: "nginx"
      volumeMounts:
        - name: pvc-disk
          mountPath: "/data"
  volumes:
  - name: pvc-disk
    persistentVolumeClaim:
      claimName: pvc-disk
```

Mount a disk as a dynamically provisioned volume

To mount a disk as a dynamically provisioned volume, you must create a PVC, create a StorageClass, and then specify the disk type in storageClassName of the PVC.

1. **Create a StorageClass**

```
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
  name: alicloud-disk-common-hangzhou-b
provisioner: alicloud/disk
parameters:
  type: cloud_ssd
  regionid: cn-hangzhou
  zoneid: cn-hangzhou-b
```

Parameters

- provisioner: Set this parameter to alicloud/disk, which indicates that the Provisioner plug-in is used to create the StorageClass.
- type: Specify the disk type. Valid values: cloud, cloud_efficiency, cloud_ssd, and available. If you set this parameter to available, the system attempts to create a disk in the following order: ultra disk, standard SSD, and basic disk, and stops until a disk is created.
- regionid: Specify the region of the disk.
- zoneid: Specify the zone of the disk.

2. **Create a Service**

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: disk-common
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: alicloud-disk-common-hangzhou-b
  resources:
    requests:
      storage: 20Gi
---
kind: Pod
apiVersion: v1
metadata:
  name: disk-pod-common
spec:
  containers:
  - name: disk-pod
    image: nginx
    volumeMounts:
      - name: disk-pvc
        mountPath: "/mnt"
  restartPolicy: "Never"
  volumes:
    - name: disk-pvc
      persistentVolumeClaim:
        claimName: disk-common
```

Default options

By default, Kubernetes clusters support the following types of StorageClass:

- alicloud-disk-common: basic disk.
- alicloud-disk-efficiency: ultra disk.
- alicloud-disk-ssd: standard SSD.
- alicloud-disk-available: This option ensures high availability. The system attempts to create an ultra disk first. If no ultra disk is available in the specified zone, the system attempts to create a standard SSD. If no standard SSD is available, the system attempts to create a basic disk.

3. **Create a multi-instance StatefulSet by using a disk**

   We recommend that you use the volumeClaimTemplates parameter. This parameter allows the system to dynamically create PVCs and PVs. PVCs are associated with PVs.

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
  - port: 80
    name: web
  clusterIP: None
  selector:
    app: nginx
---
apiVersion: apps/v1beta2
kind: StatefulSet
metadata:
  name: web
spec:
  selector:
    matchLabels:
      app: nginx
  serviceName: "nginx"
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
        - name: disk-common
          mountPath: /data
  volumeClaimTemplates:
  - metadata:
      name: disk-common
    spec:
      accessModes: [ "ReadWriteOnce" ]
      storageClassName: "alicloud-disk-common"
      resources:
        requests:
          storage: 10Gi
```

3.1.4.4.3. Use NAS volumes

This topic describes how to use Network Attached Storage (NAS) file systems in Kubernetes clusters.

NAS file systems can be mounted to Kubernetes clusters as the following volume types:

- Statically provisioned NAS volumes

  You can use statically provisioned NAS volumes in the following ways:

- Use the FlexVolume plug-in
  - Use a NAS file system as a volume
  - Use a NAS file system to create a persistent volume (PV) and a persistent volume claim (PVC)
- Use the Network File System (NFS) driver of Kubernetes.

- Dynamically provisioned NAS volumes

Prerequisites

A NAS file system is created in the NAS console and a mount target is added. The mount target is used to mount the file system to the Kubernetes cluster. The NAS file system and the Kubernetes cluster are deployed in the same Virtual Private Cloud (VPC) network.

Statically provisioned NAS volumes

You can use the FlexVolume plug-in provided by Alibaba Cloud or the NFS driver provided by Kubernetes to manage NAS file systems.

Use the FlexVolume plug-in

The FlexVolume plug-in allows you to use a NAS file system as a volume. You can also use a NAS file system to create a PV and a PVC.

> **Note**
> - NAS is a shared storage system that provides storage services for multiple pods at the same time.
> - server: the mount target of the NAS volume.
> - path: the mounted directory in the NAS file system. You can specify a subdirectory as a volume. If no subdirectory exists, the system automatically creates and mounts a subdirectory.
> - vers: the version of the NFS mounting protocol. Version 4.0 is supported.
> - mode: the access permissions on the mounted directory. If the root directory of the NAS file system is specified as the mounted directory, you cannot modify the access permissions. If the NAS file system stores a large amount of data, the mounting process may take a long time or fail. Therefore, we recommend that you do not set the mode parameter.

Use a NAS file system as a volume

Use the following `nas-deploy.yaml` file to create a pod.

```
apiVersion: v1
kind: Pod
metadata:
  name: "flexvolume-nas-example"
spec:
  containers:
    - name: "nginx"
      image: "nginx"
      volumeMounts:
        - name: "nas1"
          mountPath: "/data"
  volumes:
    - name: "nas1"
      flexVolume:
        driver: "alicloud/nas"
        options:
          server: "0cd8b4a576-grs79.cn-hangzhou.nas.aliyuncs.com"
          path: "/k8s"
          vers: "4.0"
```

Use a NAS file system to create a PV and a PVC

Step 1: Create a PV

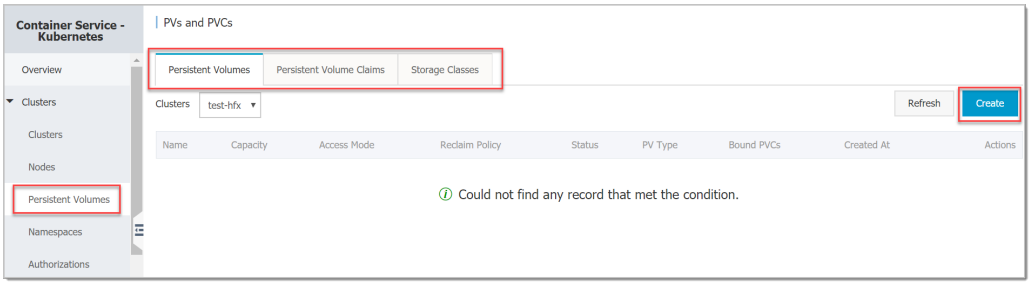
You can use a YAML file or the Container Service console to create a PV.

- Create a PV by using a YAML file

  Use the following `nas-pv.yaml` file to create a PV.

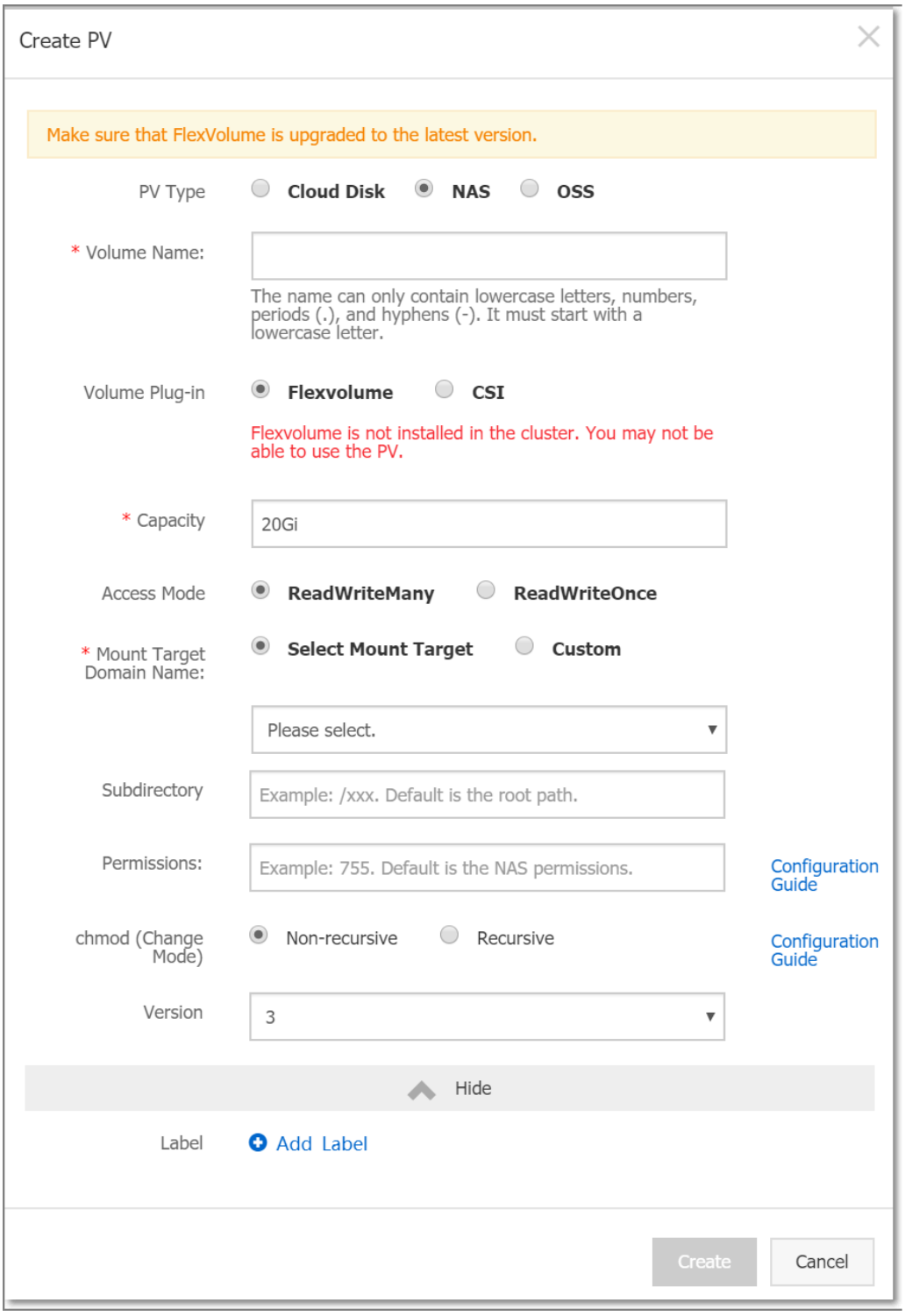
```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-nas
spec:
  capacity:
    storage: 5Gi
  storageClassName: nas
  accessModes:
    - ReadWriteMany
  flexVolume:
    driver: "alicloud/nas"
    options:
      server: "0cd8b4a576-uih75.cn-hangzhou.nas.aliyuncs.com"
      path: "/k8s"
      vers: "4.0"
```

- Create a PV in the console
  i. Log on to the Container Service console.
  ii. In the left-side navigation pane, choose **Clusters > Persistent Volumes**. The **PVs and PVCs** page appears.
  iii. On the **Persistent Volumes** tab, select the target cluster and click **Create** in the upper-right corner.



  iv. In the Create PV dialog box, set the following parameters:
    - **PV Type**: In this example, NAS is selected.
    - **Volume Name**: the name of the PV. The name must be unique in the cluster. In this example, pv-nas is specified.
    - **Volume Plug-in**: In this example, Flexvolume is selected.
    - **Capacity**: the capacity of the PV. The capacity of the PV cannot exceed that of the NAS file system.
    - **Access Mode**: The default mode is ReadWriteMany.
    - **Mount Target Domain Name**: Enter the address of the mount target that is used to mount the NAS file system to the cluster.
    - **Subdirectory**: Enter a subdirectory in the NAS file system. The subdirectory must start with a forward slash (/). If this parameter is set, the specified subdirectory is mounted as the PV.
      - If the specified subdirectory does not exist, the system automatically creates this subdirectory.
      - This parameter is optional. By default, the root directory of the NAS file system is mounted.

- **Permissions**: Set the access permissions on the mounted directory, for example, 755, 644, or 777.
  - The permissions can be set only when a subdirectory is mounted as the PV.
  - This parameter is optional. By default, the original permissions are used.
- **chmod (Change Mode)**: In this example, Non-recursive is selected.
- **Version**: The version of the NFS mounting protocol. Version 3 and 4.0 are supported. We recommend that you use the default version 3.
- **Label**: Add labels for the PV.



v. Click **Create**.

Step 2: Create a PVC

Use the following *nas-pvc.yaml* file to create a PVC.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-nas
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: nas
  resources:
    requests:
      storage: 5Gi
```

Step 3: Create a pod

Use the following *nas-pod.yaml* file to create a pod.

```
apiVersion: v1
kind: Pod
metadata:
  name: "flexvolume-nas-example"
spec:
  containers:
    - name: "nginx"
      image: "nginx"
      volumeMounts:
          - name: pvc-nas
            mountPath: "/data"
  volumes:
  - name: pvc-nas
    persistentVolumeClaim:
        claimName: pvc-nas
```

Use the NFS driver

Step 1: Create a NAS file system

Log on to the NAS console. For more information, see the *Create a file system* chapter *of* NAS User Guide.

> **Note** The NAS file system and the Kubernetes cluster must be deployed in the same region.

In this example, the following mount target is used: `055f84ad83-ixxxx.cn-hangzhou.nas.aliyuncs.com`.

Step 2: Create a PV

You can use a YAML template or the Container Service console to create a PV.

- **Create a PV by using a YAML template**

  Use the *nas-pv.yaml* file to create a PV.

  Run the following command to create a PV from the NAS file system:

```
root@master # cat << EOF |kubectl apply -f -
apiVersion: v1
kind: PersistentVolume
metadata:
   name: nas
spec:
   capacity:
     storage: 8Gi
   accessModes:
     - ReadWriteMany
   persistentVolumeReclaimPolicy: Retain
   nfs:
     path: /
     server: 055f84ad83-ixxxx.cn-hangzhou.nas.aliyuncs.com
EOF
```

- **Create a PV in the console**

  For more information, see Use a NAS file system to create a PV and a PVC.

Step 3: Create a PVC

Create a PVC and associate the PVC with the PV that is created in Step 2.

```
root@master # cat << EOF | kubectl apply -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
     name: nasclaim
spec:
     accessModes:
       - ReadWriteMany
     resources:
       requests:
         storage: 8Gi
EOF
```

Step 4: Create a Pod

Create an application to use the PV.

```
root@master # cat << EOF |kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
     name: mypod
spec:
     containers:
       - name: myfrontend
         image: registry.aliyuncs.com/spacexnice/netdia:latest
         volumeMounts:
         - mountPath: "/var/www/html"
           name: mypd
     volumes:
       - name: mypd
         persistentVolumeClaim:
           claimName: nasclaim
EOF
```

The NAS file system is now mounted to the application that runs in the pod.

Dynamically provisioned NAS volumes

If you want to use dynamically provisioned NAS volumes, install a driver plug-in and configure a NAS mount target.

Install the plug-in

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: alicloud-nas
provisioner: alicloud/nas
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: alicloud-nas-controller
  namespace: kube-system
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: run-alicloud-nas-controller
subjects:
  - kind: ServiceAccount
    name: alicloud-nas-controller
    namespace: kube-system
roleRef:
  kind: ClusterRole
  name: alicloud-disk-controller-runner
  apiGroup: rbac.authorization.k8s.io
---
kind: Deployment
apiVersion: extensions/v1beta1
metadata:
  name: alicloud-nas-controller
  namespace: kube-system
spec:
  replicas: 1
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: alicloud-nas-controller
    spec:
      tolerations:
      - effect: NoSchedule
        operator: Exists
        key: node-role.kubernetes.io/master
      - effect: NoSchedule
        operator: Exists
        key: node.cloudprovider.kubernetes.io/uninitialized
      nodeSelector:
         node-role.kubernetes.io/master: ""
      serviceAccount: alicloud-nas-controller
      containers:
        - name: alicloud-nas-controller
          image: registry.cn-hangzhou.aliyuncs.com/acs/alicloud-nas-controller:v1.8.4
          volumeMounts:
```

```
          - mountPath: /persistentvolumes
            name: nfs-client-root
          env:
            - name: PROVISIONER_NAME
              value: alicloud/nas
            - name: NFS_SERVER
              value: 0cd8b4a576-mmi32.cn-hangzhou.nas.aliyuncs.com
            - name: NFS_PATH
              value: /
      volumes:
        - name: nfs-client-root
          nfs:
            server: 0cd8b4a576-mmi32.cn-hangzhou.nas.aliyuncs.com
            path: /
```

Use a dynamically provisioned NAS volume

```
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: web
spec:
  serviceName: "nginx"
  replicas: 2
  volumeClaimTemplates:
  - metadata:
      name: html
    spec:
      accessModes:
        - ReadWriteOnce
      storageClassName: alicloud-nas
      resources:
        requests:
          storage: 2Gi
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:alpine
        volumeMounts:
        - mountPath: "/usr/share/nginx/html/"
          name: html
```

3.1.4.4.4. Mount an OSS bucket to a cluster

You can mount Object Storage Service (OSS) buckets as volumes in Kubernetes clusters.

You can mount OSS buckets in the following ways:

- Mount an OSS bucket as a volume.
- Create a PV and a PVC.

Prerequisites

To mount an OSS bucket as a statically provisioned volume, you must create an OSS bucket first.

Background information

- OSS buckets can be mounted as only statically provisioned volumes.
- An OSS bucket can be mounted to one or more pods.
- bucket: Only buckets can be mounted to a Kubernetes cluster. The subdirectories or files in a bucket cannot be mounted to a Kubernetes cluster.
- url: Specify the endpoint of the OSS bucket. The endpoint is the domain name that is used to mount the OSS bucket to the Kubernetes cluster.
- akId: Specify your AccessKey ID.
- akSecret: Specify your AccessKey secret.
- otherOpts: the custom parameters that are used to mount the OSS bucket. The parameters must be in the following format: `-o *** -o ***`.

> **Note** To mount an OSS bucket as a volume, you must create a Secret with your AccessKey information when you deploy the flexvolume Service.

Mount an OSS bucket as a statically provisioned volume

- **Mount an OSS bucket as a volume**

  Use the following *oss-deploy.yaml* file to create a pod:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-oss-deploy
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx-flexvolume-oss
        image: nginx
        volumeMounts:
          - name: "oss1"
            mountPath: "/data"
      volumes:
        - name: "oss1"
          flexVolume:
            driver: "alicloud/oss"
            options:
              bucket: "docker"
              url: "oss-cn-hangzhou.aliyuncs.com"
              akId: ***
              akSecret: ***
              otherOpts: "-o max_stat_cache_size=0 -o allow_other"
```

- **Create a static PV and a PVC**

  i. **Create a PV**

  You can create a PV in the Container Service console or by using a YAML file.

- **Create a PV by using a YAML file**

  Use the following *oss-pv.yaml* file to create a PV:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-oss
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteMany
  storageClassName: oss
  flexVolume:
    driver: "alicloud/oss"
    options:
      bucket: "docker"
      url: "oss-cn-hangzhou.aliyuncs.com"
      akId: ***
      akSecret: ***
      otherOpts: "-o max_stat_cache_size=0 -o allow_other"
```

- **Create a PV in the Container Service console**

  a. Log on to the Container Service console.

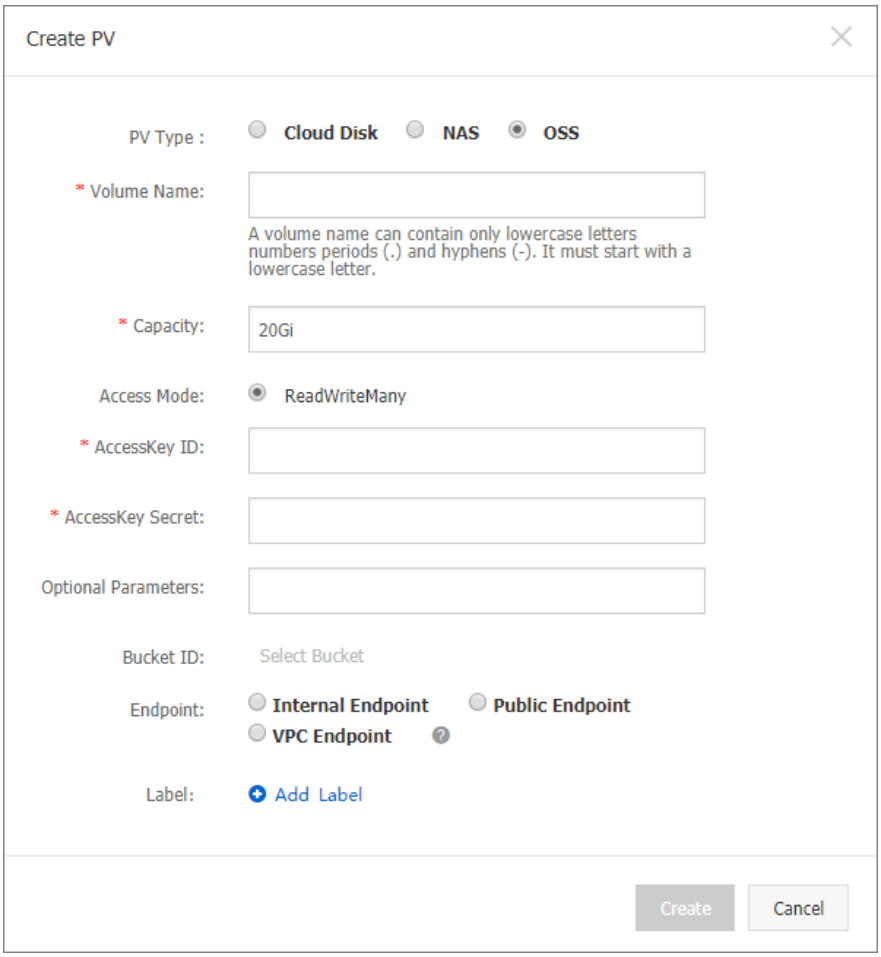
  b. In the left-side navigation pane, choose **Clusters > Persistent Volumes** to go to the **Persistent Volumes** page.

  c. On the **Persistent Volumes** tab, select the cluster that you want to manage and click **Create** in the upper-right corner.

  d. Set the parameters in the Create PV dialog box. Create a PV in the console

Parameter	Description
PV Type	In this example, OSS is selected.
Volume Name	Enter a name for the PV. The name must be unique in the cluster. In this example, pv-oss is used.
Volume Plug-in	You can select Flexvolume or CSI.
Capacity	Specify the capacity of the PV.
Access Mode	By default, ReadWriteMany is selected.
AccessKey ID and **AccessKey Secret**	The AccessKey pair is required to access OSS buckets. To obtain your AccessKey pair, go to the Apsara Uni-manager Management Console, choose **Enterprise > Organizations**, click  on the right side of the organization. Then, click **AccessKey** and copy the AccessKey pair.
Optional Parameters	Enter custom parameters in the format of `-o ** * -o ***`.

Parameter	Description
Bucket ID	Enter the name of the OSS bucket that you want to mount. Click **Select Bucket**. In the dialog box that appears, select the OSS bucket that you want to mount and click **Select**.
Endpoint	We recommend that you choose **Internal Endpoint**.
Label	Add labels to the PV.



e. Click **Create**.

ii. **Create a PVC**

Use the following *oss-pvc.yaml* file to create a PVC:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-oss
spec:
  storageClassName: oss
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 5Gi
```

iii. **Create a pod**

Use the following *oss-pod.yaml* file to create a pod:

```
apiVersion: v1
kind: Pod
metadata:
  name: "flexvolume-oss-example"
spec:
  containers:
    - name: "nginx"
      image: "nginx"
      volumeMounts:
          - name: pvc-oss
            mountPath: "/data"
  volumes:
  - name: pvc-oss
    persistentVolumeClaim:
        claimName: pvc-oss
```

Can I mount an OSS bucket as a dynamically provisioned volume?

No, you cannot mount an OSS bucket as a dynamically provisioned volume.

3.1.4.4.5. Create a PVC

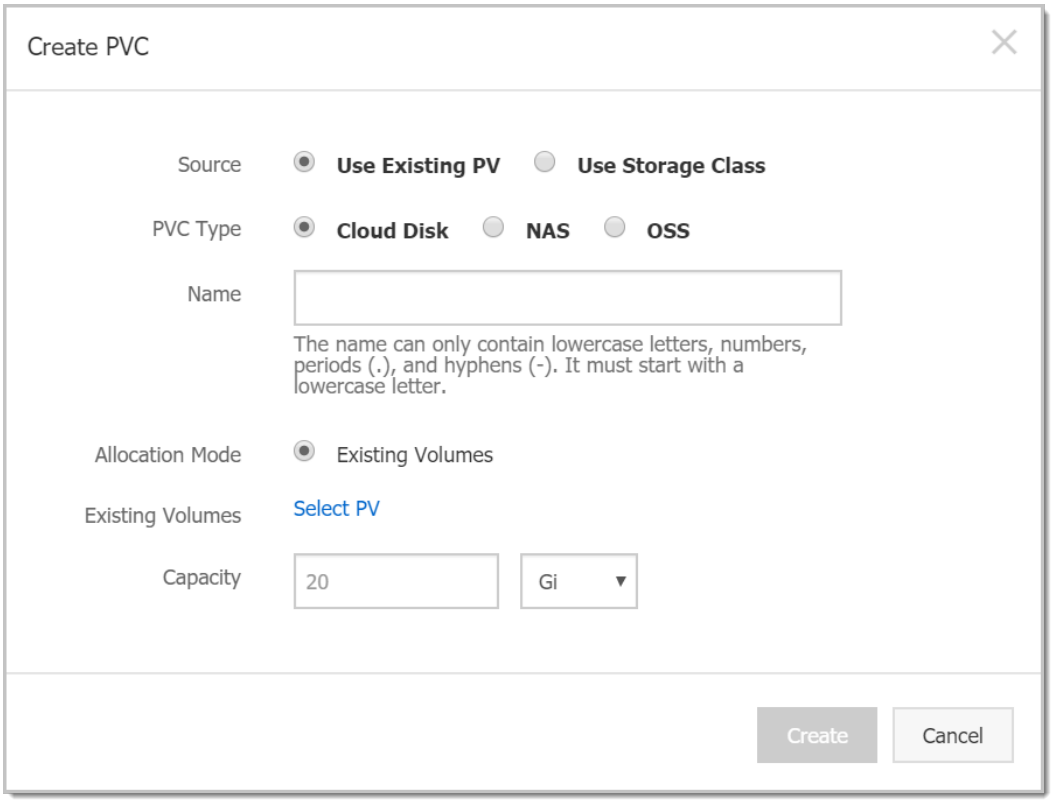
Prerequisites

- You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.
- You have created a PV. This example uses a PV created from a disk. For more information, see Use Apsara Stack disks.

  By default, PVCs are associated with PVs that have the alicloud-pvname label. PVs created through the Container Service console all have this label. If a PV does not have this label, you need to add the label before you can associate it with a PVC.

Procedure

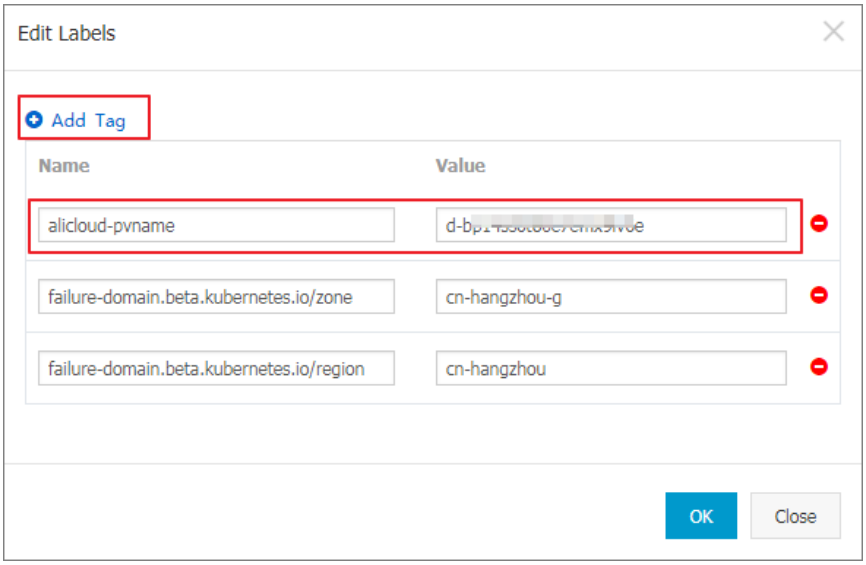
1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Applications > Persistent Volume Claims**. The **PVs and PVCs** page appears.
3. On the **Persistent Volume Claims** tab, select the target cluster and namespace, and click **Create** in the upper-right corner.
4. In the Create PVC dialog box, set the parameters, and click **Create**.



- **Source**: Select an existing PV or use a storage class.
- **PVC Type**: The same as the types of PVs. Three types, cloud disk, NAS, and OSS, are supported.
- **Name**: The name of the PVC.
- **Allocation Mode**: Currently, only existing volumes are supported.
- **Existing Volumes**: Select PVs of the same type as the PVC.
- **Capacity**: The claimed usage, which cannot be larger than the total capacity of associated PVs.

> **Note** If your cluster has a PV that is not used, but you cannot find it in the **Select PV** dialog box, the reason may be that the PV does not have the alicloud-pvname label.

If you cannot find available PVs, you can choose **Clusters > Persistent Volumes** in the left-side navigation pane. Find the PV that you want to use and click **Manage Labels** in the Actions column. You can attach a label to the PV and set the label name to alicloud-pvname and the value to the PV name. By default, the disk ID is used as the PV name if the PV is created from a disk.



5. On the Persistent Volume Claims page, the newly created PVC is now displayed.

3.1.4.4.6. Use a PVC

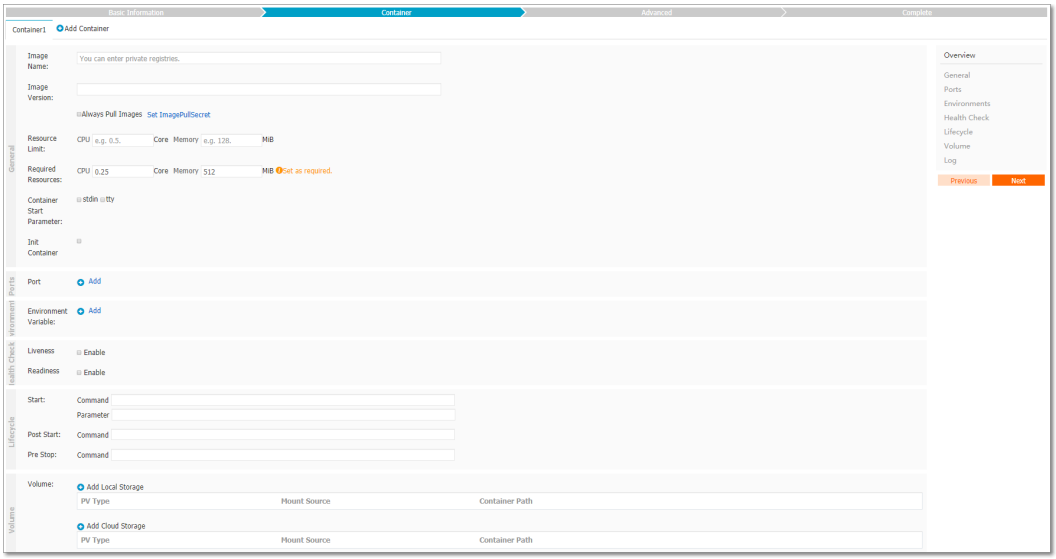
You can use persistent volume claims (PVCs) in your applications.

Prerequisites

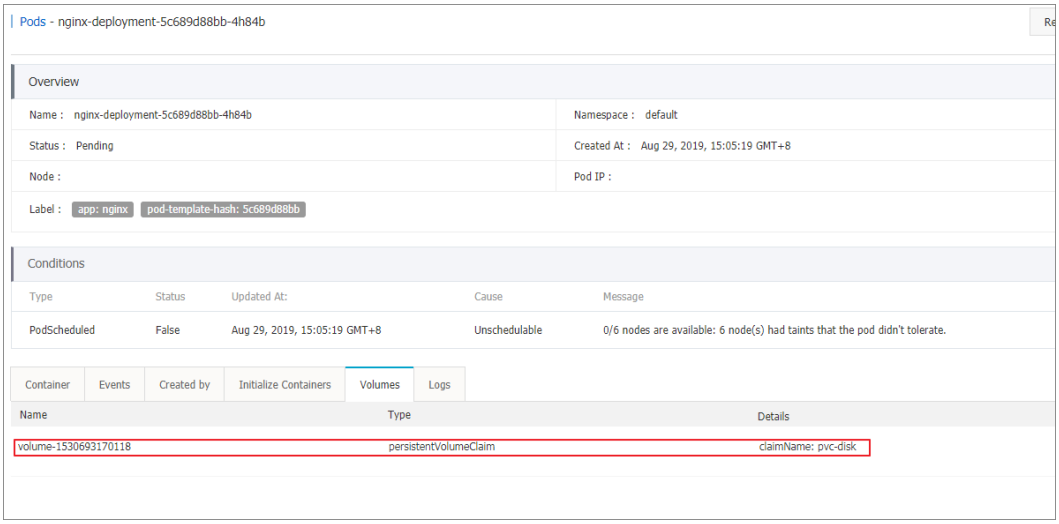
- You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.
- You have created a PVC. This example uses a PVC named pvc-disk. For more information, see Create PVCs.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Applications > Deployments**. The **Deployments** page appears. In the upper-right corner, click **Create from Image**.
3. On the Basic Information page, specify the application name, cluster, namespace, number of replicas, type, labels, and annotations. Then click **Next**.
4. On the Container page, select the image. Then, specify the type of cloud volume. Currently, cloud disks, NAS, and OSS are supported. In this example, select the pvc-disk PVC and click **Next**.



5. Configure the test-nginx application, and then click **Create**.
6. After the application is created, choose **Applications > Pods** in the left-side navigation pane. Select the pod to which the application belongs, and click **View Details**.
7. On the pod details page, click the **Volumes** tab. Verify that the pod is now associated with the pvc-disk PVC.



3.1.4.5. Network management

3.1.4.5.1. Set access control for pods

This topic describes how to use network policies to control access between pods.

Prerequisites

You have created a Kubernetes cluster and selected the **Terwaynetwork plug-in**. For more information, see Create a Kubernetes cluster.

Context

You can declare network policies to control access between pods and thus prevent applications from interfering each other.

Procedure

For more information about standard Kubernetes network policies, see Network policies.

1. Create a pod that runs as a server and attach `label run=nginx` to the pod. For more information, see Create an application from an orchestration template.

   The sample YAML file is as follows:

```
apiVersion: v1
kind: Pod
metadata:
  name: server
  labels:
    run: nginx
spec:
  containers:
    - name: nginx
      image: registry.acs.intranet.env22.com/nginx:1.8
```

2. Create a network policy. For more information, see Create an application from an orchestration template.

   The sample YAML file is as follows:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: access-nginx
spec:
  podSelector:
    matchLabels:
      run: nginx # Apply the network policy to pods with the run=nginx label
  ingress:
  - from:
    - podSelector:
        matchLabels:
          access: "true" # Only pods with the access=true label are accessible
```

3. Use the *client.yaml* and *client-label* files to create two pods that run as clients.

   One pod has the required label and the other does not.

   i. Create the *client.yaml* and *client-label* files with the following contents respectively.

```
# This pod has no label
apiVersion: v1
kind: Pod
metadata:
  name: client
spec:
  containers:
    - name: busybox
      image: registry.acs.intranet.env22.com/acs/busybox
      command: ["sh", "-c", "sleep 200000"]
```

```
# This pod has the label
apiVersion: v1
kind: Pod
metadata:
  name: client-label
  labels:
    access: "true"
spec:
  containers:
    - name: busybox
      image: registry.acs.intranet.env22.com/acs/busybox
      command: ["sh", "-c", "sleep 200000"]
```

   ii. Run the following commands to create these pods:

```
kubectl apply -f client.yaml
kubectl apply -f client-label.yaml
```

   You can see that only the pod with the required label can access the server.

3.1.4.5.2. Set bandwidth limits for pods

This topic describes how to limit the bandwidth of inbound and outbound traffic that flows through a pod.

Prerequisites

You have created a Kubernetes cluster and selected the **Terwaynetwork plug-in**. For more information, see

Create a Kubernetes cluster.

Context

Throttling pods helps prevent performance degradation of the host or other workloads when certain pods occupy excessive resources.

Method

You can use the `k8s.aliyun.com/ingress-bandwidth` and `k8s.aliyun.com/egress-bandwidth` annotations for pod throttling.

- `k8s.aliyun.com/ingress-bandwidth`: limits the pod inbound bandwidth.
- `k8s.aliyun.com/egress-bandwidth`: limits the pod outbound bandwidth.
- The bandwidth limit is measured in m and k, which represent Mbit/s and Kbit/s respectively.

Procedure

1. Create a pod that runs as a server in the console. For more information, see Create an application from an orchestration template.

   The sample YAML file is as follows:

   ```
   apiVersion: v1
   kind: Pod
   metadata:
     name: server
     annotations:
       k8s.aliyun.com/ingress-bandwidth: 10m # Set the inbound bandwidth limit to 10 Mbit/s
       k8s.aliyun.com/egress-bandwidth: 10m
   spec:
     containers:
       - name: nginx
         image: registry.acs.intranet.env22.com/nginx:1.8
   ```

2. Run the kubectl exec command to connect to the pod. To verify that pod throttling is effective, run the following commands to create a file on the pod. Assume that the IP address of the pod created in step 1 is 172.16.XX.XX.

   ```
   cd /usr/share/nginx/html
   dd if=/dev/zero of=bigfile bs=1M count=1000
   ```

3. Use the *client-deploy.yaml* file to create a pod that runs as a client.

   i. Create the *client-deploy.yaml* file with the following content:

   ```
   apiVersion: v1
   kind: Pod
   metadata:
     name: client
     annotations:
        k8s.aliyun.com/ingress-bandwidth: 10m # Set the inbound bandwidth limit to 10 Mbit/s
        k8s.aliyun.com/egress-bandwidth: 10m
   spec:
     containers:
       - name: busybox
         image: registry.acs.intranet.env22.com/acs/netdia
         command: ["sh", "-c", "sleep 200000"]
   ```

   ii. Run the following command to create the pod:

   ```
   kubectl apply -f client-deploy.yaml
   ```

4. Run the following command to check whether bandwidth is limited:

```
kubectl exec -it client sh
```

3.1.4.6. Namespaces

3.1.4.6.1. Create a namespace

You can create a namespace in the console.

Prerequisites

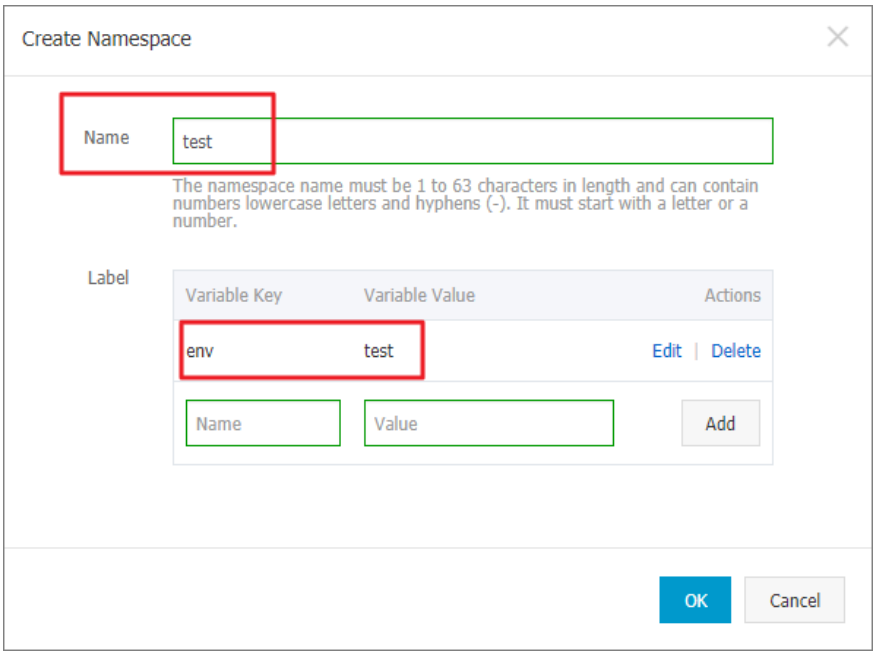
You have created a Kubernetes cluster.

Context

In a Kubernetes cluster, you can use namespaces to create multiple virtual spaces. When a large number of users share a cluster, you can use namespaces to divide different workspaces and allocate cluster resources to different tasks. Furthermore, you can use resource quotas to allocate resources to each namespace.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Namespaces**. The Namespaces page appears.
3. Select the target cluster and click **Create** in the upper-right corner.
4. In the dialog box that appears, set the parameters.

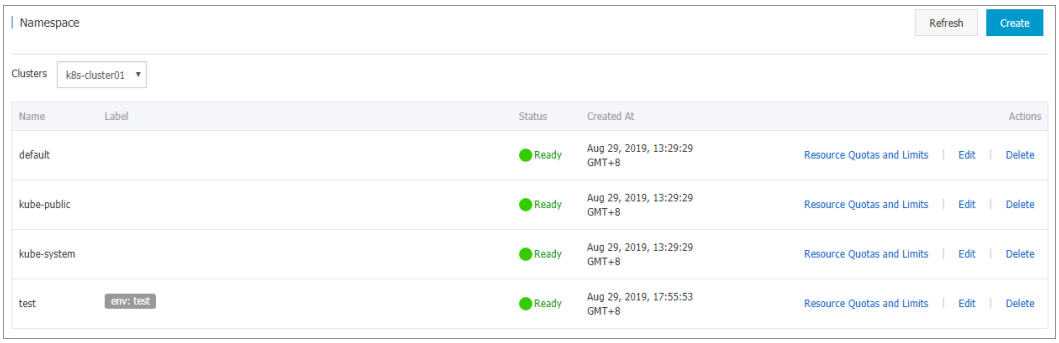


Create a namespace

Parameter	Description
Name	Enter a name for the namespace. In this example, enter test. The name must be 1 to 63 characters in length and can contain digits, letters, and hyphens (-). It must start and end with a letter or digit.

Parameter	Description
Label	**Label**: Add one or more labels to the namespace. Labels are used to add marks to namespaces. For example, you can use a label to mark the namespace as one used in the test environment. Enter a variable key and value. Then click **Add** on the right to add a label.

5. Click **OK**.
6. The newly created namespace is now displayed on the Namespaces page.



3.1.4.6.2. Set resource quotas and limits

You can set resource quotas and limits for a namespace in the console.

Prerequisites

- You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.
- You have created the sample namespace **test**. For more information, see Create a namespace.
- You can connect to a master node of the cluster. For more information, see Connect to a Kubernetes cluster through kubectl.

Context

By default, a running pod uses the CPU and memory resources of a node without limit. This means that any pod can use the computing resources of a cluster without restraints. Therefore, pods of a namespace may deplete the cluster resources.

Namespaces can be used as virtual clusters to serve multiple purposes and meet different needs. We recommend that you set resource quotas for all namespaces.

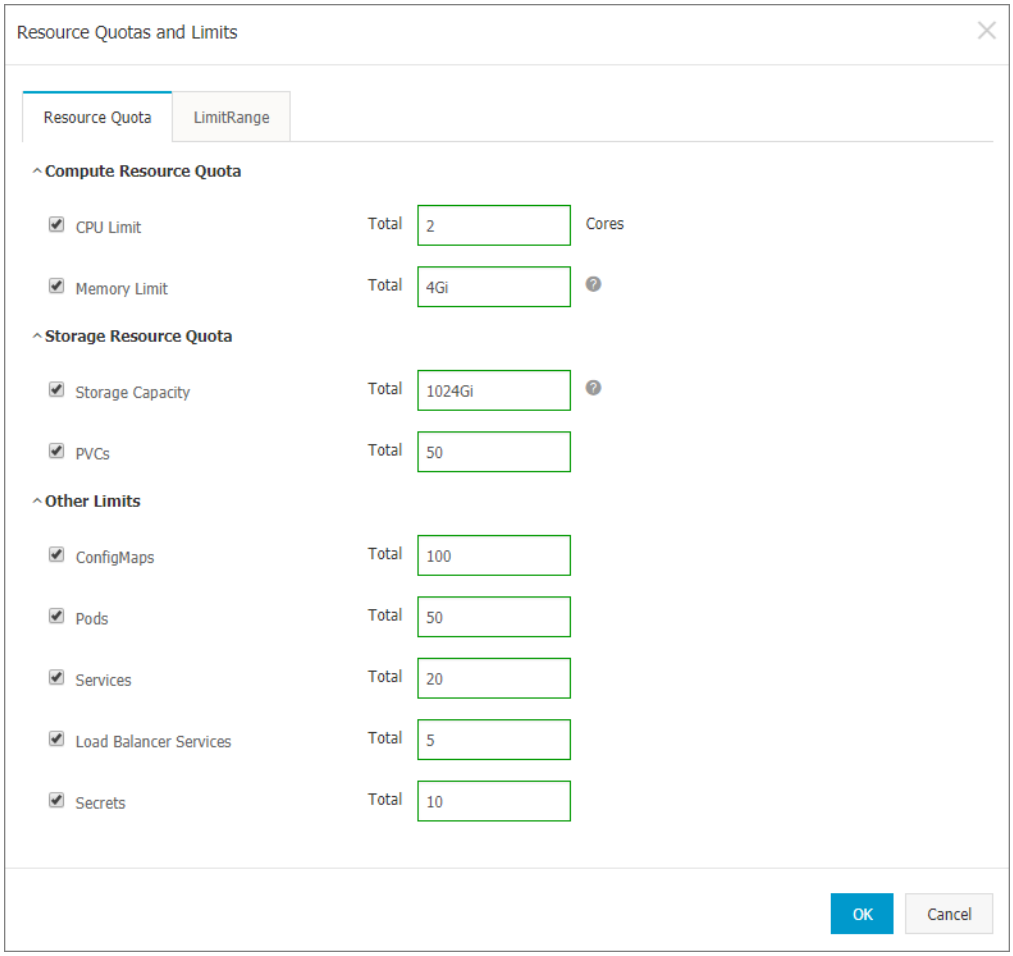
For a namespace, you can set quotas on resources such as CPU, memory, and pod quantity. For more information, see Resource Quotas.

Procedure

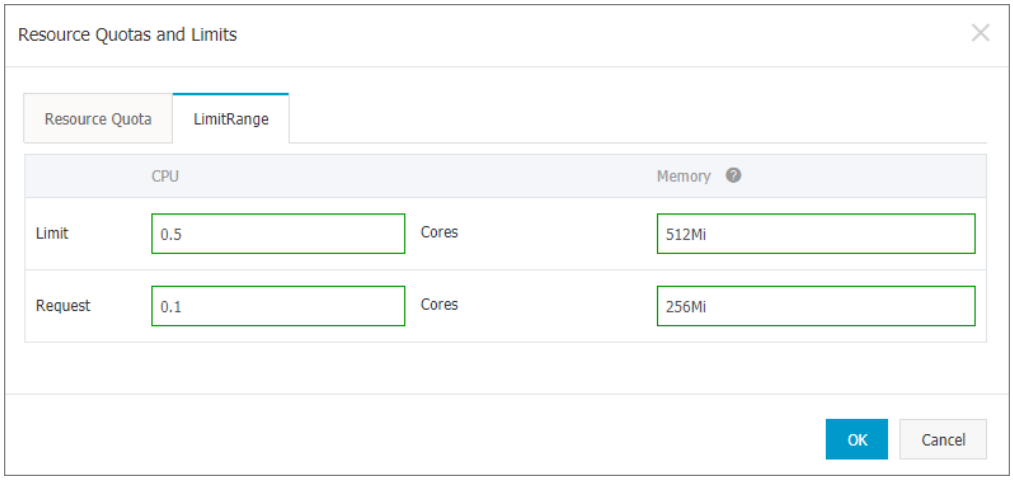
1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Namespaces**. On the Namespaces page, select the target cluster. Find the target namespace and click **Resource Quotas and Limits** in the Actions column.
3. In the dialog box that appears, set resource quotas and default resource limits.

> **Note** After you set CPU and memory quotas for a namespace, you must specify CPU and memory limits when you create a pod. Alternatively, you can set the default resource limits for the namespace. For more information, see Resource Quotas.

i. Set resource quotas for the namespace.



ii. You can set resource limits and resource requests for containers in the namespace. This enables you to control the amount of resources consumed by containers. For more information, see https://kubernetes.io//memory-default-namespace/.



4. After you set resource quotas and limits, connect to a master node of the cluster and run the following commands to query the resource configurations of the namespace.

```
# kubectl get limitrange,ResourceQuota -n test
NAME AGE
limitrange/limits 8m
NAME AGE
resourcequota/quota 8m
# kubectl describe limitrange/limits resourcequota/quota -n test
Name: limits
Namespace: test
Type Resource Min Max Default Request Default Limit Max Limit/Request Ratio
---- -------- --- --- --------------- ------------- -----------------------
Container cpu - - 100m 500m -
Container memory - - 256Mi 512Mi -
Name: quota
Namespace: test
Resource Used Hard
-------- ---- ----
configmaps 0 100
limits.cpu 0 2
limits.memory 0 4Gi
persistentvolumeclaims 0 50
pods 0 50
requests.storage 0 1Ti
secrets 1 10
services 0 20
services.loadbalancers 0 5
```

3.1.4.6.3. Edit a namespace

You can edit an existing namespace.

Prerequisites

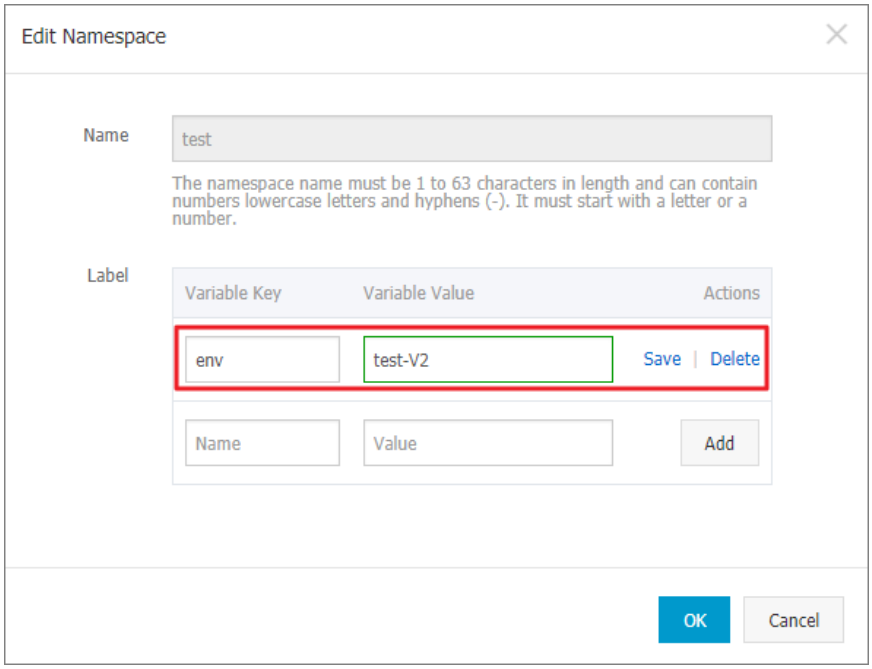
- You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.
- You have created the sample namespace **test**. For more information, see Create a namespace.

Context

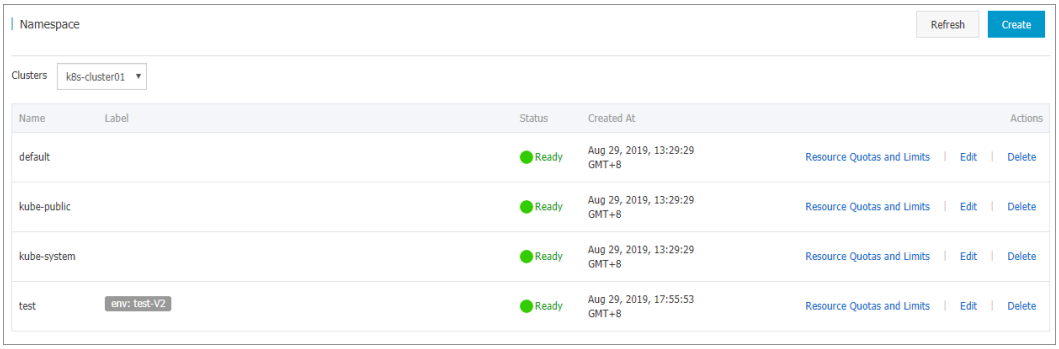
When you edit a namespace, you can add, delete, or modify namespace labels based on needs.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Namespaces**. The Namespaces page appears.
3. Select the target cluster. Find the namespace that you want to edit and click **Edit** in the Actions column.
4. In the dialog box that appears, select a label and click **Edit** to modify its key and value. This example changes a label to `env:test-V2`. Then click **Save**.



5. Click **OK**. Go to the Namespaces page and check the newly edited namespace label.



3.1.4.6.4. Delete a namespace

You can delete namespaces that are no longer in use.

Prerequisites

- You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.
- You have created the sample namespace **test**. For more information, see Create a namespace.

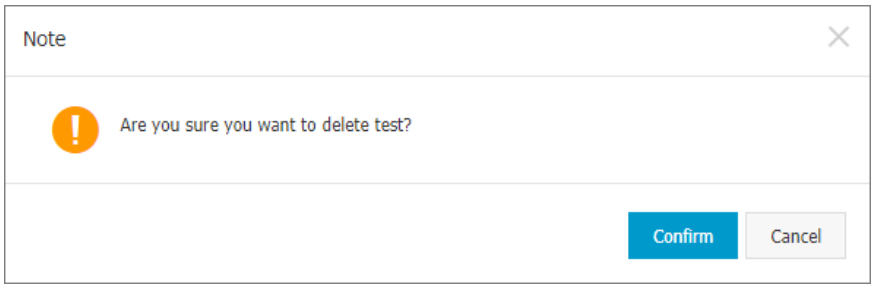
Context

Note When you delete a namespace, all resource objects under the namespace will be deleted. Exercise caution when you perform this operation.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Namespaces**. The Namespaces page appears.
3. Select the target cluster. Find the namespace that you want to delete and click **Delete** in the Actions column.

4. In the dialog box that appears, click **Confirm**.



5. The namespace is now deleted from the Namespaces page. Resource objects under the namespace are also deleted.

3.1.4.7. Applications

3.1.4.7.1. Create an application from an image

You can use an image to create an NGINX application that is accessible over the Internet.

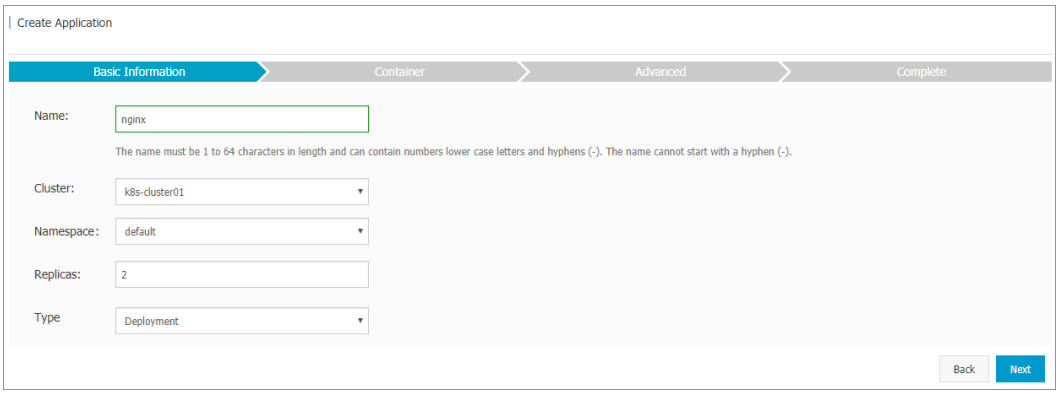
Prerequisites

- A Kubernetes cluster is created. For more information, see Create a Kubernetes cluster.
- Your Kubernetes cluster is accessible over the Internet.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Applications > Deployments** to go to the Deployments page. Then, click **Create from Image** in the upper-right corner of the page.
3. In the dialog box that appears, set the parameters, including **Name**, **Cluster**, **Namespace**, **Replicas**, and **Type**, and select **Synchronize Timezone**. Click **Next**.

   If you do not set the **Namespace** parameter, the default namespace is used.



4. Configure containers.

> **Note** You can configure multiple containers for the pods of the application.

i. Configure general settings.

Container general settings

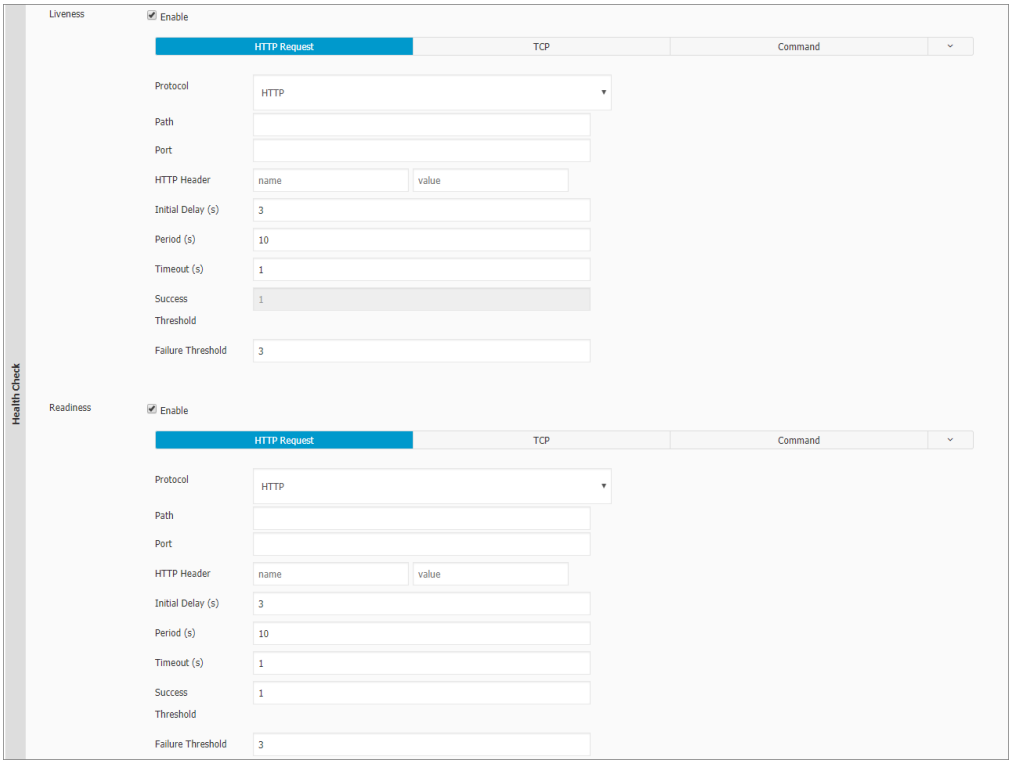
Parameter	Description
Image Name	You can click **Select Image**, and in the dialog box that appears, select the required image and click **OK**. In this example, NGINX is selected. You can also enter a private registry URL to specify the image. The registry URL follows this format: `domainname/namespace/imagename`.
Image Version	You can click **Select Image Version** to select the required version. If you do not specify the image version, the latest version is used.
Always Pull Images	To improve efficiency, Container Service caches images. During the deployment, if the version of the specified image is the same as that of a cached image, Container Service will reuse the cached image instead of pulling the image again. Therefore, when you update the application code, if you do not change the image version for reasons such as to support the upper-layer workloads, the previously cached image will be used. When this check box is selected, Container Service will always pull the image from the repository to deploy the application. This ensures that the latest image and code are used.
Set Image Pull Secret	Click Set Image Pull Secret to set the secret. The secret is required if you need to access a private repository.
Resource Limit	The upper limits of CPU and memory resources that are available to this application. This prevents the application from occupying excessive resources. The unit of CPU resources is Core. The unit of memory is MiB.
Required Resources	The amount of CPU and memory resources that are reserved for this application. These resources are exclusive to the container. This prevents the application from being unavailable if other services or processes share the resources.
Container Start Parameter	Select **stdin** to enable standard inputs for the container. Select **tty** to assign a virtual terminal that is used to send signals to the container. We recommend that you select both check boxes. This allows you to associate the terminal (tty) with the standard inputs (stdin) of the container. For example, an interactive program can be used to obtain standard inputs from users and then display the inputs on the terminal.
Init Container	When this check box is selected, the system creates an Init Container that contains useful tools. For more information, see .

ii. (Optional)Set environment variables.

You can use key-value pairs to set environment variables for pods. Environment variables are used to apply pod configurations to containers. For more information, see Pod variable.

iii. (Optional)Configure health check settings.

Health check settings include liveness and readiness probes. Liveness probes determine when to restart the container. Readiness probes determine whether the container is ready to start accepting traffic. For more information about health checks, see .



Request type	Description
HTTP request	Sends an HTTP GET request to the container. Supported parameters include: ■ Protocol: HTTP or HTTPS. ■ Path: the requested path on the server. ■ Port: the port opened in the container. The port number must range from 1 to 65535. ■ HTTP Header: The custom headers in the HTTP request. Duplicate headers are allowed. Key-value pairs are supported. ■ Initial Delay (s): the **initialDelaySeconds** field, which specifies the period between when the container is started and when the system performs the first probe. Unit: seconds. Default value: 3. ■ Period (s): the **periodSeconds** field, which specifies the intervals between two adjacent probes. Unit: seconds. Default value: 10. Minimum value: 1. ■ Timeout (s): the **timeoutSeconds** field, which specifies the period after which a probe times out. Unit: seconds. Default value: 1. Minimum value: 1. ■ Healthy Threshold: the minimum number of consecutive successes that must occur for a probe to be considered successful after having failed. Default value: 1. Minimum value: 1. For liveness probes, this parameter must be set to 1. ■ Unhealthy Threshold: the minimum number of consecutive failures that must occur for a probe to be considered failed after having succeeded. Default value: 3. Minimum value: 1.

Request type	Description
TCP connection	Sends a TCP socket to the container. The kubelet will attempt to open the socket on the specified port. If the connection can be established, the container is considered healthy. Otherwise, it is considered unhealthy. Supported parameters include: ■ Port: the port opened in the container. The port number must range from 1 to 65535. ■ Initial Delay (s): the **initialDelaySeconds** field, which specifies the period between when the container is started and when the system performs the first probe. Unit: seconds. Default value: 15. ■ Period (s): the **periodSeconds** field, which specifies the intervals between two adjacent probes. Unit: seconds. Default value: 10. Minimum value: 1. ■ Timeout (s): the **timeoutSeconds** field, which specifies the period after which a probe times out. Unit: seconds. Default value: 1. Minimum value: 1. ■ Healthy Threshold: the minimum number of consecutive successes that must occur for a probe to be considered successful after having failed. Default value: 1. Minimum value: 1. For liveness probes, this parameter must be set to 1. ■ Unhealthy Threshold: the minimum number of consecutive failures that must occur for a probe to be considered failed after having succeeded. Default value: 3. Minimum value: 1.
Command line	Runs a probe command in the container to check its health status. Supported parameters include: ■ Command: the probe command that is used to check the health status of the container. ■ Initial Delay (s): the **initialDelaySeconds** field, which specifies the period between when the container is started and when the system performs the first probe. Unit: seconds. Default value: 15. ■ Period (s): the **periodSeconds** field, which specifies the intervals between two adjacent probes. Unit: seconds. Default value: 10. Minimum value: 1. ■ Timeout (s): the **timeoutSeconds** field, which specifies the period after which a probe times out. Unit: seconds. Default value: 1. Minimum value: 1. ■ Healthy Threshold: the minimum number of consecutive successes that must occur for a probe to be considered successful after having failed. Default value: 1. Minimum value: 1. For liveness probes, this parameter must be set to 1. ■ Unhealthy Threshold: the minimum number of consecutive failures that must occur for a probe to be considered failed after having succeeded. Default value: 3. Minimum value: 1.

iv. Configure lifecycle events.

You can set the following parameters to configure the lifecycle of the container in the Start, Post Start, and Pre Stop fields. For more information, visit https://kubernetes.io/docs/tasks/configure-pod-container/attach-handler-lifecycle-event/.

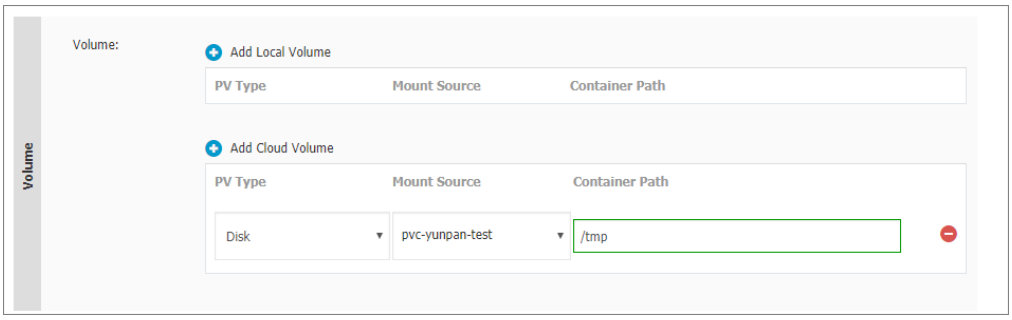
- **Start**: the pre-start command and parameter.
- **Post Start**: the post-start command.
- **Pre Stop**: the pre-stop command.

v. (Optional)Configure volumes.

Local storage and cloud storage are supported.

- **Local Storage**: supports hostPaths, ConfigMaps, secrets, and temporary directories, and mounts a mount source to the specified path in the container. For more information, see Volumes.
- **Cloud Storage**: supports three types of persistent volumes (PVs): cloud disks, Network Attached Storage (NAS), and Object Storage Service (OSS).

In this example, a PV is created from a cloud disk, and the PV is mounted to the */tmp* path in the container. Data generated in this path is stored in the cloud disk.



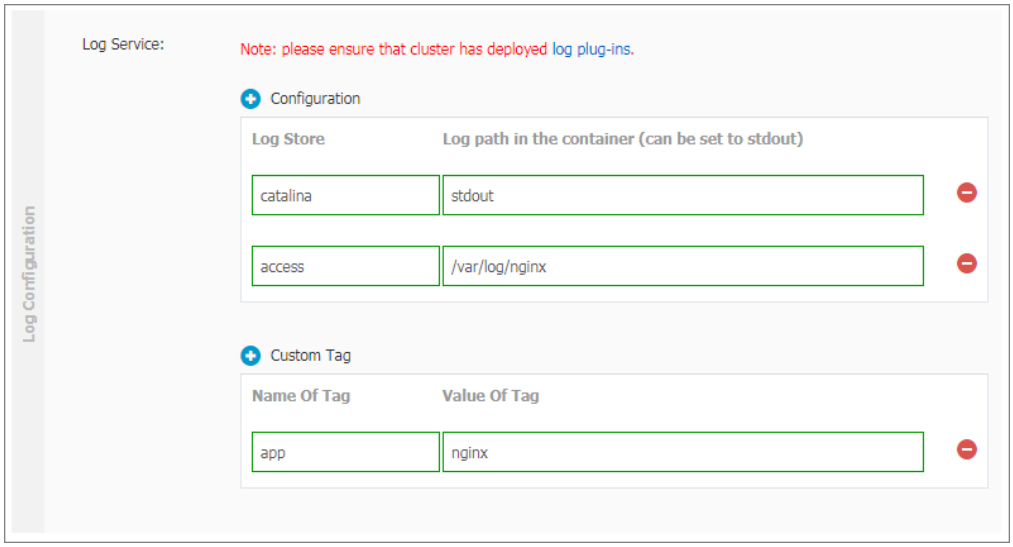
vi. (Optional)Configure **Log Service**. You can configure collection methods and custom tags.

> **Note** Make sure that the Log Service agent is installed in the cluster.

Log collection parameters include:

- **Logstore**: the Logstore that is used to store log data in Log Service.
- **Log Path in Container**: You can set this parameter to stdout or a log path.
  - **stdout**: collects standard outputs of the container.
  - **Log Path**: collects logs in the specified path of the container. In this example, logs in the following path are collected: */var/log/nginx*. Wildcards are supported.

You can also set custom tags, which will be collected and output along with logs. Custom tags simplify statistical analysis of log data.

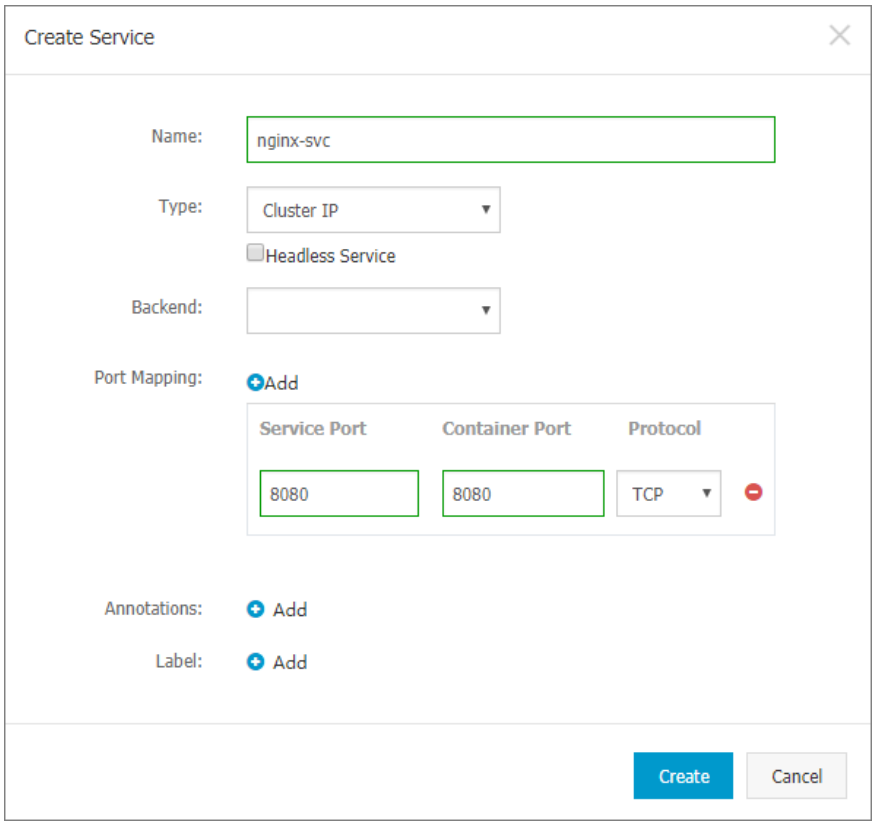


5. Set other parameters based on your needs and click **Next**.
6. Configure advanced settings, including the access control settings.

   You can configure the method to access pods and click **Create** to create the application. In this example, a service of the Cluster IP type and an Ingress are created to enable Internet access to the NGINX application.

> **Note**
>
> You can configure access control settings based on your needs:
>
> - Internal applications: For applications that run inside the cluster, you can create a service of the Cluster IP or Node Port type to enable internal communication to fit your needs.
> - External applications: For applications that need to be accessed over the Internet, you can configure access control settings by using one of the following methods:
>   - Create a service of the Server Load Balancer (SLB) type and enable access to your application over the Internet by using the SLB instance.
>   - Create a service of the Cluster IP or Node Port type, create an Ingress, and then enable access to your application over the Internet by using the Ingress. For more information, see .

i. To create a service, click **Create** in the **Service** section. In the dialog box that appears, configure the service and click **Create**.

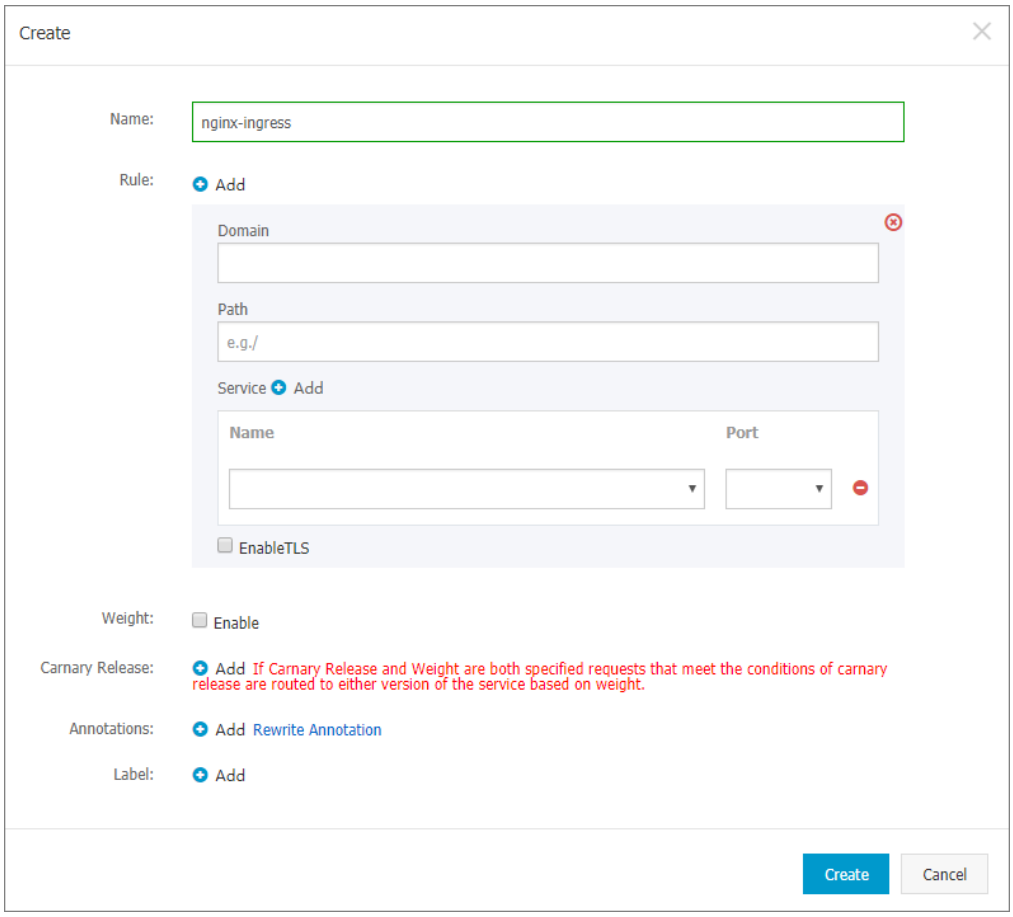


Parameter	Description
Name	Enter a name for the service. Default value: `applicationname-svc`.
Type	Select one of the following types: ■ Cluster IP: enables access to the service through an internal IP address of the cluster. If you select this type, the service is only accessible within the cluster. ■ Node Port: enables access to the service through the IP address and static port on each node. The NodePort field specifies the static port. A NodePort service can be used to route requests to a Cluster IP service. The system automatically creates the Cluster IP service. You can access a Node Port service from outside the cluster by requesting `<NodeIP>:<NodePort>`. ■ Server Load Balancer: enables access to the service by using an SLB instance over the Internet or an internal network. The SLB instance can route requests to Node Port and Cluster IP services.
Port Mapping	Set a service port and a container port. If the **Type** parameter is set to Node Port, you must set a node port to avoid port conflicts. TCP and UDP protocols are supported.
Annotations	Add annotations to the service. SLB parameters are supported. For more information, see Access services by using SLB.
Label	Add labels to the service.

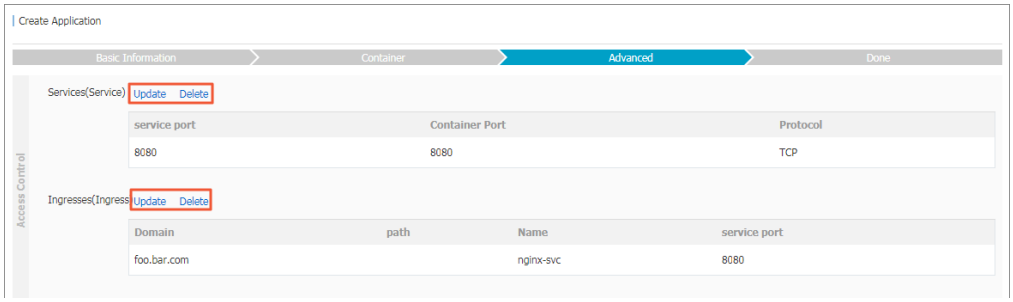
ii. To create an Ingress, click **Create** in the **Ingress** section. In the dialog box that appears, configure Ingress rules and click **Create**. For more information about Ingress configuration, see Ingress configurations.

When you create an application from an image, you can create an Ingress for only one service. In this example, a virtual host name is used as the test domain. You must add the following entry to the hosts file to map the domain to the IP address of the Ingress. In actual scenarios, use a domain that has obtained an ICP number.

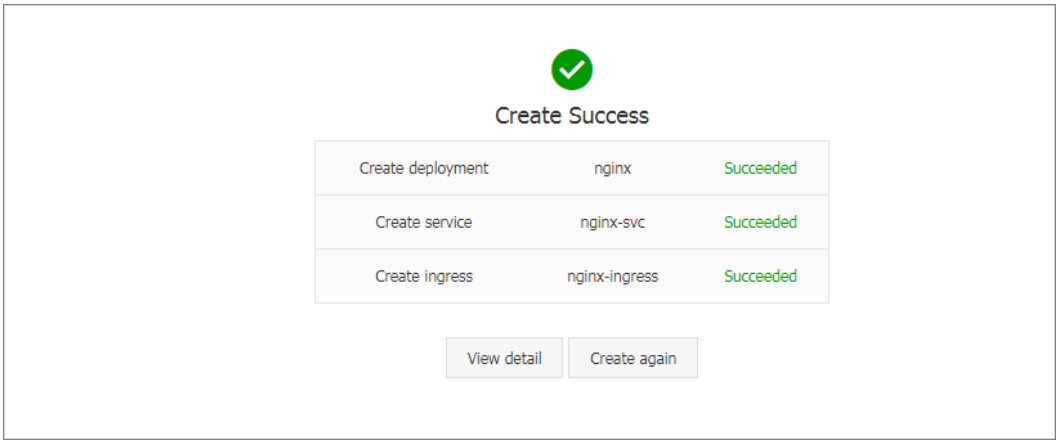
```
101.37.224.146   foo.bar.com    #The IP address of the Ingress.
```



iii. You can find the newly created service and Ingress in the Access Control section. You can click **Update** or **Delete** to make changes.

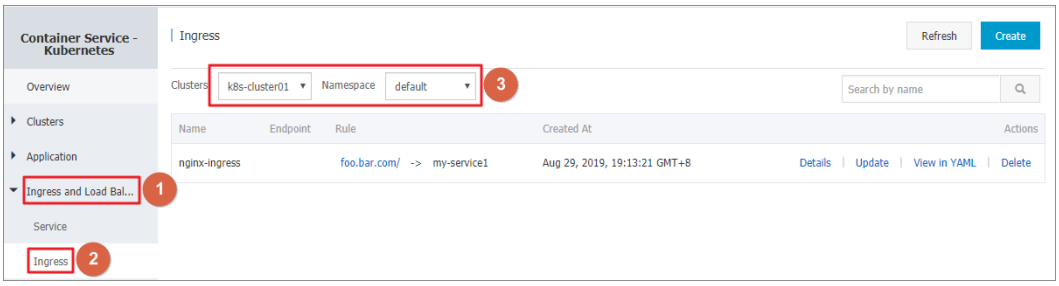


7. Click **Create**.
8. After the application is created, a message appears to display the resource objects included in the application. You can click **View Details** to view application details.



The nginx-deployment page is displayed by default.

9. In the left-side navigation pane, choose **Ingresses and Load Balancing > Ingresses**. The following rule is displayed on the Ingresses page.



10. Enter the test domain in the address bar of your browser and press the Enter key. The NGINX welcome page appears.

3.1.4.7.2. Create an application from an orchestration template

Container Service provides orchestration templates that you can use to create applications quickly. You can also modify the templates based on YAML syntax to customize applications.

Prerequisites

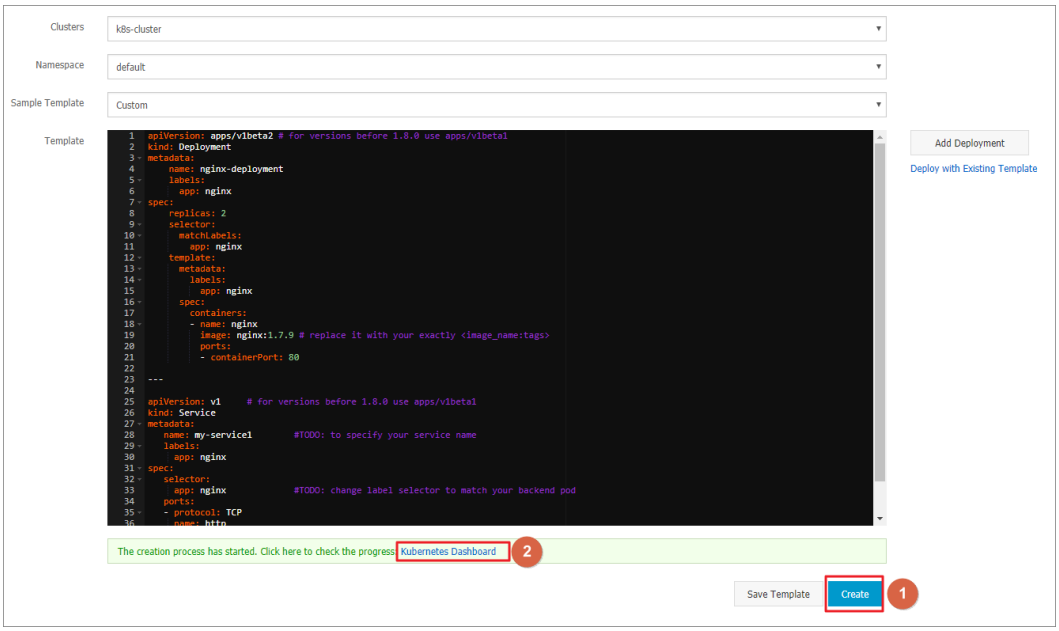
You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.

Context

The following example demonstrates how to create an NGINX application consisting of a deployment and a service. The service is associated with a pod created by the deployment.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Applications > Deployments**. The **Deployments** page appears.
3. In the upper-right corner, click **Create from Template**.
4. Set the parameters and click **Create**.
   - **Cluster**: Select the cluster where the resource objects are to be deployed.
   - **Namespace**: Select the namespace to which the resource objects belong. The default namespace is default. Except for underlying computing resources such as nodes and PVs, most resources are scoped to namespaces.
   - **Sample Template**: Container Service provides YAML templates of various resource types to help you deploy resource objects quickly. You can also create a custom template based on YAML syntax to describe the resource that you want to define.
   - **Add Deployment**: This feature allows you to quickly define a YAML template.
   - **Use Existing Template**: You can import an existing template to the configuration page.



Based on an orchestration template provided by Container Service, the following sample template creates a deployment of an NGINX application.

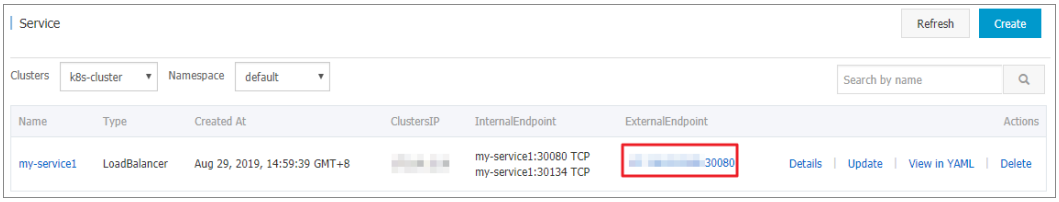
> **Note** Container Service supports YAML syntax. You can use the `---` symbol to separate multiple resource objects. This enables you to create multiple resource objects in a single template.

```
apiVersion: apps/v1beta2 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
    name: nginx-deployment
    labels:
      app: nginx
spec:
    replicas: 2
    selector:
      matchLabels:
        app: nginx
    template:
      metadata:
        labels:
          app: nginx
      spec:
        containers:
        - name: nginx
          image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
          ports:
          - containerPort: 80
---
apiVersion: v1     # for versions before 1.8.0 use apps/v1beta1
kind: Service
metadata:
   name: my-service1        #TODO: to specify your service name
   labels:
     app: nginx
spec:
   selector:
     app: nginx             #TODO: change label selector to match your backend pod
   ports:
   - protocol: TCP
     name: http
     port: 30080            #TODO: choose an unique port on each node to avoid port conflict
     targetPort: 80
   type: LoadBalancer        ## This example changes the service type from NodePort to LoadBalanc
er.
```

5. Click **Create**. A message appears indicating the deployment status.

   In the left-side navigation pane, choose **Ingresses and Load Balancing > Services** to view the newly created service.

6. On Kubernetes Dashboard, verify that a my-service1 service is running and its external endpoint is displayed. Click the address in the **External Endpoint** column.



7. You can visit the NGINX welcome page in the browser.

What's next

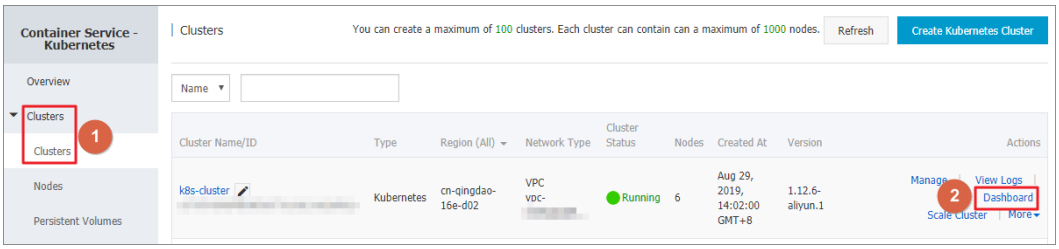
You can also choose **Ingresses and Load Balancing > Services** in the left-side navigation pane to view the NGINX service.

3.1.4.7.3. Create an application through Kubernetes Dashboard

You can create an application through Kubernetes Dashboard.

Procedure

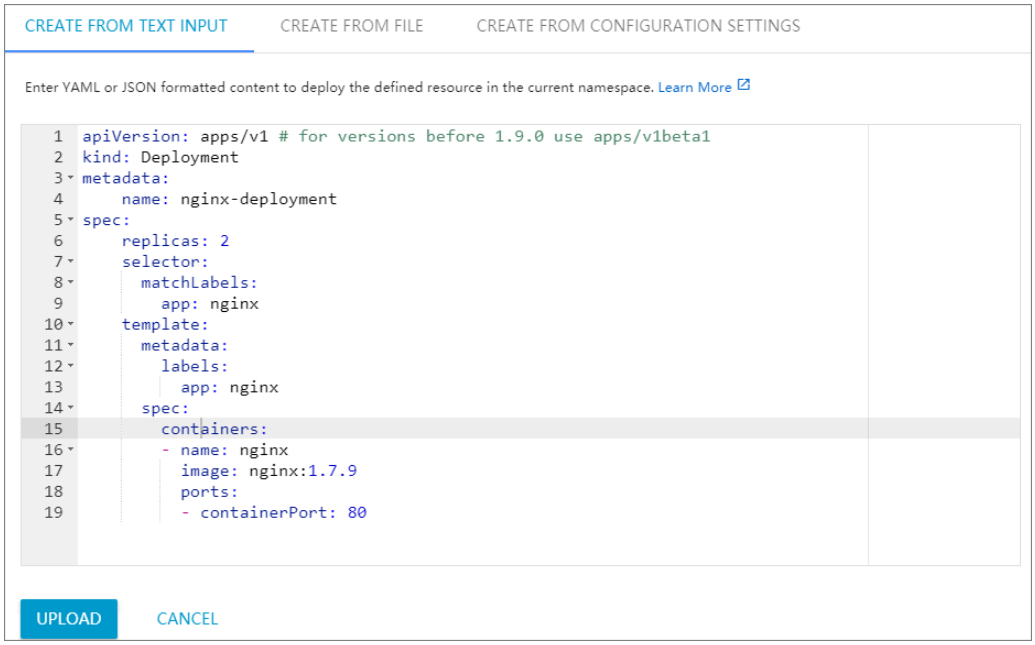
1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters**. The Clusters page appears.
3. Find the target cluster and click **Dashboard** in the Actions column to go to Kubernetes Dashboard.



4. On the Overview page, click **CREATE** in the upper-right corner.
5. In the dialog box that appears, configure the application.

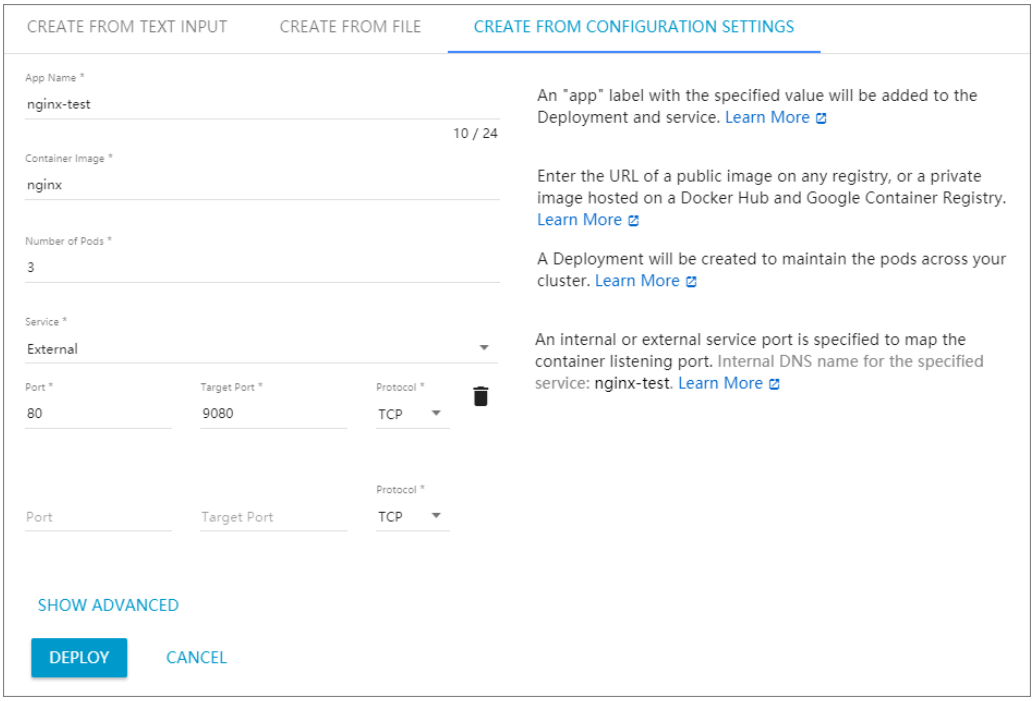
   You can use one of the following methods to create the application.

   - **CREATE FROM TEXT INPUT**: Directly enter an orchestration template in YAML or JSON format to create the application. A sample template in YAML format is provided as follows:



- **CREATE FROM FILE**: Import a YAML or JSON configuration file to create the application.
- **CREATE AN APP**: Set parameters to create the application. Application parameters

Parameter	Description
App name	The name of the application. In this example, enter `nginx`.
Container image	The URL of the image. In this example, a Docker NGINX image is used.
Number of pods	The number of pods that constitute the application.
Service	Valid values: **External** and **Internal**. **External**: Create a service that is accessible from outside the cluster. **Internal**: Create a service that is accessible within the cluster.
Advanced options	Click **SHOW ADVANCED OPTIONS** to set labels and environment variables. The following setting distributes traffic to three pods based on load balancing.



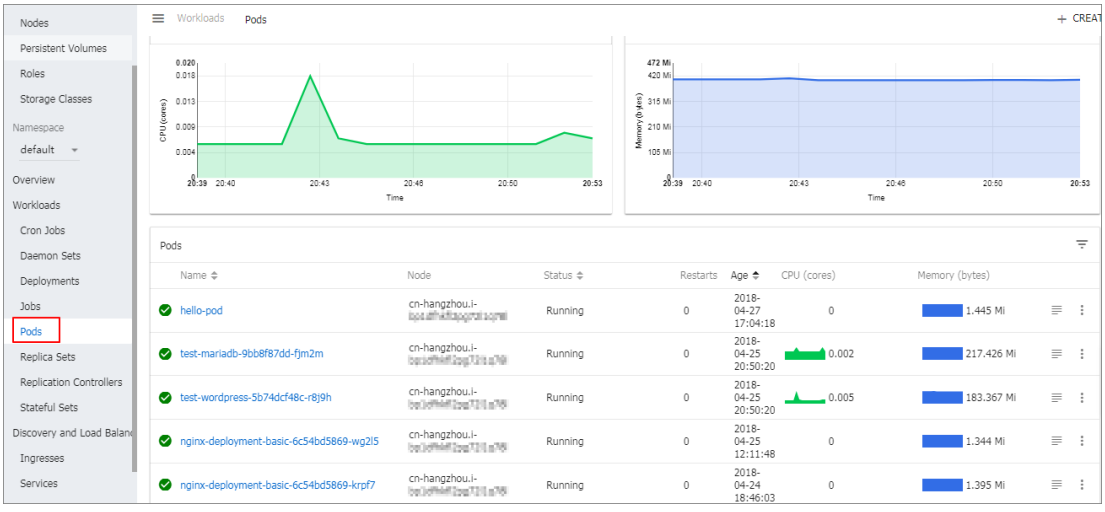
6. Click **DEPLOY** to deploy the pods and service.

   You can also click **SHOW ADVANCED OPTIONS** to configure other parameters.

What's next

After you click **DEPLOY**, you can click the left-side navigation pane to view the service or pods that constitute the application.

In the left-side navigation pane, click **Pods**. The icon on the left side of each pod indicates the status of the pod. indicates that the pod is being deployed. indicates that the pod has been successfully deployed. indicates that an error occurred while deploying the pod.



3.1.4.7.4. Use commands to manage applications

You can use commands to create applications or view application containers.

Prerequisites

Before you use commands on your local host, you have connected to a Kubernetes cluster through kubectl. For more information, see Connect to a Kubernetes cluster through kubectl.

Run a command to create an application

You can use the following command to run a simple container (an NGINX Web server in this example):

```
# kubectl run -it nginx --image=registry.aliyuncs.com/spacexnice/netdia:latest
```

This command creates a service portal for this container. After you specify `--type=LoadBalancer`, an SLB route to the NGINX container is created.

```
# kubectl expose deployment nginx --port=80 --target-port=80 --type=LoadBalancer
```

Run a command to view container information

Run the following command to list all running containers in the default namespace:

```
root@master # kubectl get pods
    NAME                                READY     STATUS    RESTARTS   AGE
    nginx-2721357637-dvwq3              1/1       Running   1          9h
```

3.1.4.7.5. Create a service

You can create a service for your application through the Container Service console. The service provides access to the application.

A Kubernetes service, generally known as microservice, is an abstraction which defines a logical set of pods and a policy by which to access the pods. A label selector usually determines whether the set of pods can be accessed by the service.

Each pod has its own IP address. Pods are created and deleted dynamically and quickly. Using pods to provide services externally is therefore not a highly available solution. The service abstraction enables the decoupling between the frontend and the backend. The frontend does not need to be aware of how the backend is implemented, which leads to a loosely coupled microservices based architecture.

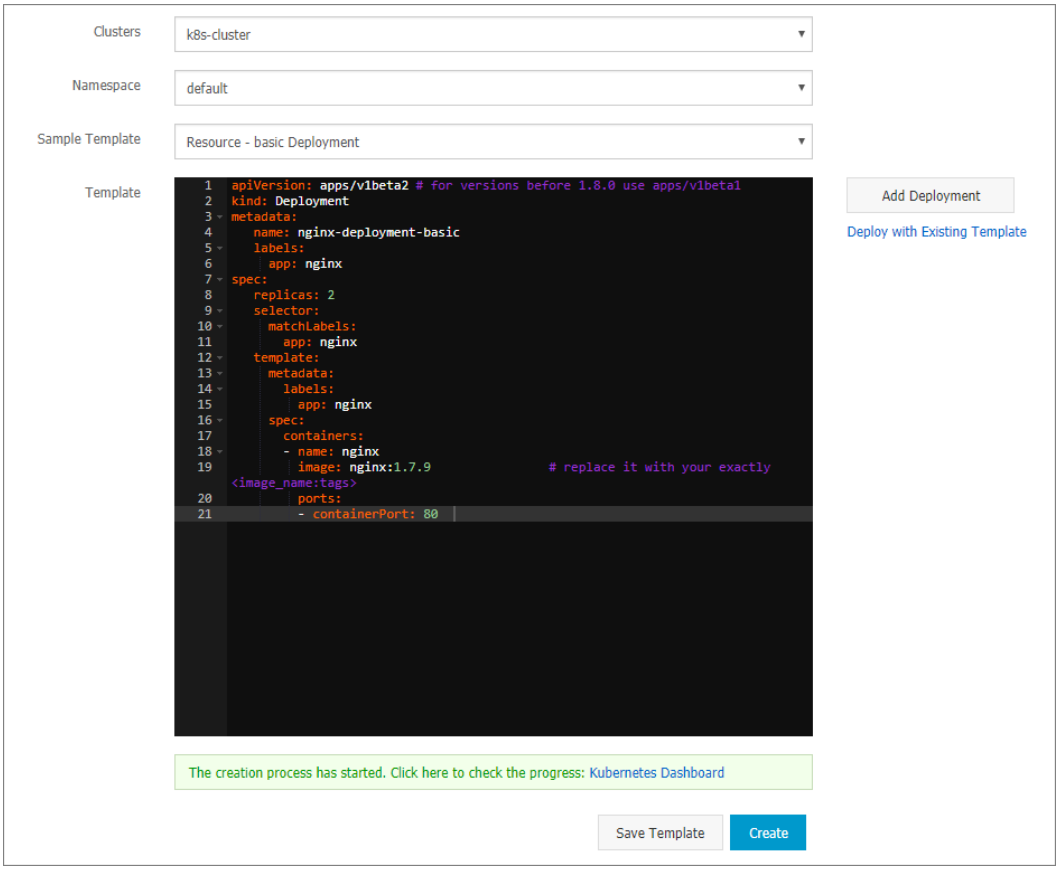
For more information, see Kubernetes service.

Prerequisites

You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.

Step 1: Create a deployment

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Applications > Deployments**. In the upper-right corner, click **Create from Template**.
3. Select the target cluster and namespace. Set the Sample Template field to Custom and enter the following code in the Template field. Then click **Create**.



In this example, the template of an NGINX deployment is used.

```
apiVersion: apps/v1beta2 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
   name: nginx-deployment-basic
   labels:
     app: nginx
spec:
   replicas: 2
   selector:
     matchLabels:
       app: nginx
   template:
     metadata:
       labels:
         app: nginx
     spec:
       containers:
       - name: nginx
         image: nginx:1.7.9               # replace it with your exactly <image_name:tags>
         ports:
         - containerPort: 80              ##expose this port in the service
```

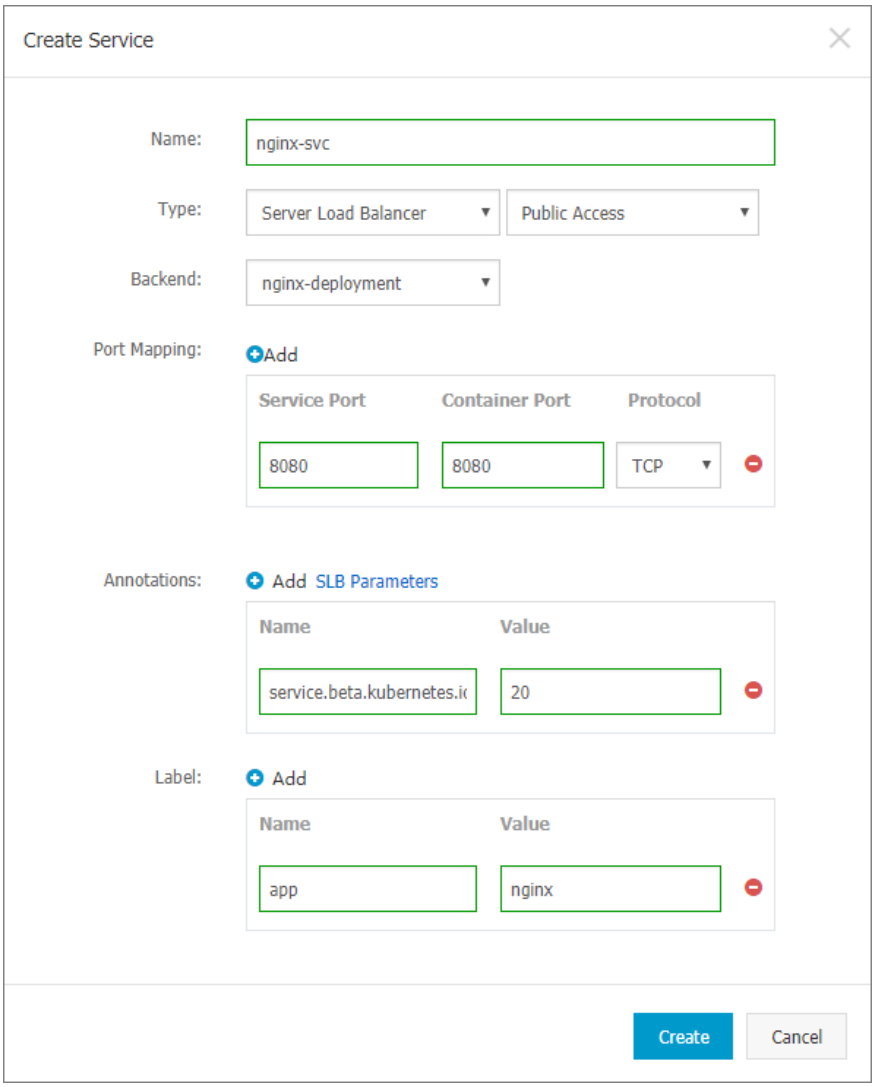
4. Click **Kubernetes Dashboard** to view the status of the deployment.

Step 2: Create a service

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Ingresses and Load Balancing > Services**. The Services page

appears.

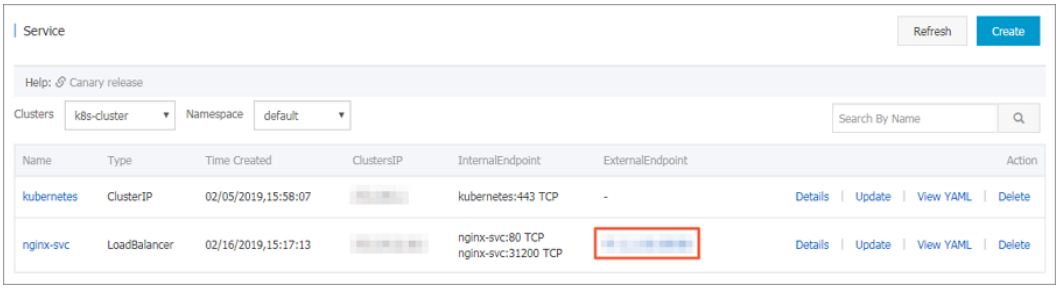
3. Select the target cluster and namespace. Then click **Create** in the upper-right corner.
4. In the Create Service dialog box that appears, set the following parameters.



- **Name**: The name of the service. In this example, enter nginx-svc.
- **Type**: The type of the service, namely, how to expose the service.
  - Cluster IP: Exposes the service through an internal IP address in the cluster. When this option is selected, the service is only accessible within the cluster. This is the default service type.
  - Node Port: Exposes the service through the IP address and static port (NodePort) of each node. A NodePort service can route requests to a Cluster IP service, which is automatically created by the system. You can access a Node Port service from outside the cluster by requesting `<NodeIP>:<NodePort>`.
  - Server Load Balancer: Exposes the service through an SLB instance, which supports Internet access or internal access. An SLB service can route requests to Node Port and Cluster IP services.
- **Backend**: The backend object that you want to associate with the service. In this example, select nginx-deployment-basic created from the previous step. If you do not specify a deployment, no Endpoint object will be created. You can manually bind the service to an Endpoint object. For more information, see services-without-selectors.
- **Port Mapping**: Set the service port and container port. The container port must be the same as the one exposed by the backend pod.

- **Annotations**: Add one or more annotations to the service to configure SLB parameters. For example, set the name to `service.beta.kubernetes.io` and value to `20`. This means that the maximum bandwidth of the service is 20 Mbit/s. For more information, see Access services by using SLB.
- **Label**: Add labels to the service.

5. Click **Create**. Service nginx-svc is displayed on the Services page.
6. You can view basic information about the service. You can also access its external endpoint through a browser.



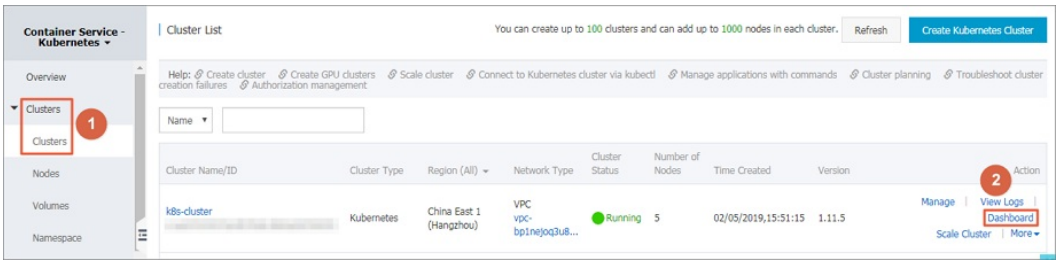
You have created a service and associated it with a backend deployment. You can now visit the NGINX welcome page.

3.1.4.7.6. Scale a service

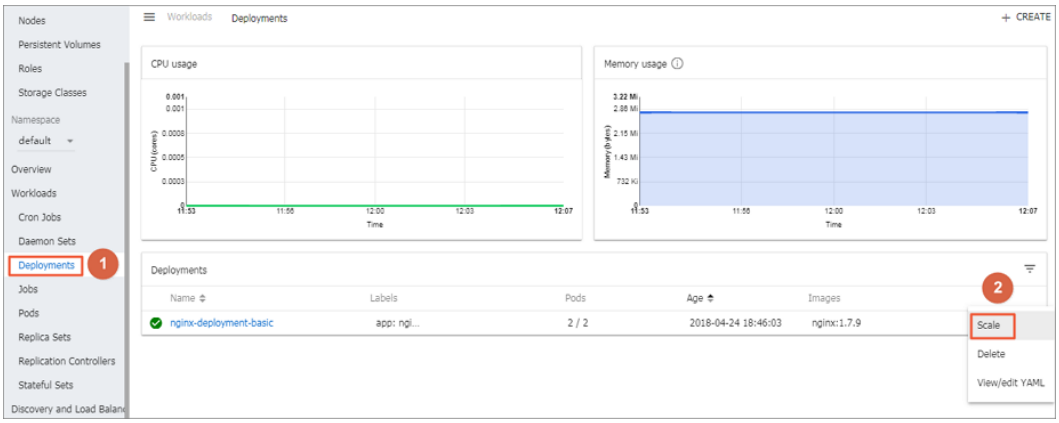
After an application is created, you can scale in or scale out the service based on your needs.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters**. The Clusters page appears.
3. Find the target cluster and click **Dashboard** in the Actions column to go to Kubernetes Dashboard.

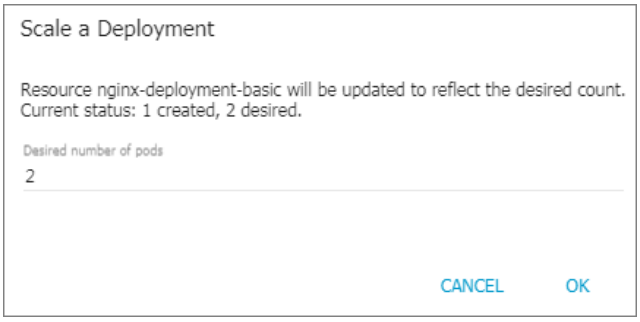


4. In the left-side navigation pane, click **Deployments**.
5. Find the target application, click the More icon on the right, and choose **Scale**.



6. In the dialog box that appears, change the **Desired number of pods** to 2. Then, click **OK**.

   This action adds a new pod and increases the number of replicas to two.



What's next

The icon on the left side of each Kubernetes object indicates the status of the object. indicates that the object is being deployed. indicates that the object has been successfully deployed.

After a deployment is complete, you can click the deployment name to view details of the running web services. You can view the replica sets included in the deployment, and the CPU and memory usage of these replica sets. You can also click to view container logs.

> **Note** If no resources are displayed, wait a few minutes and then refresh the page.

3.1.4.7.7. View a service

You can view details about a service through the Container Service console.

Context

If an external service is configured when you create an application, Kubernetes Dashboard creates the external service and preconfigures the SLB instance to direct traffic to containers in the cluster.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Ingresses and Load Balancing > Services**. The Services page appears.
3. Select the target cluster and namespace. Find the target service and click **Details** in the Actions column.
4. (Optional)You can also go to Kubernetes Dashboard. In the left-side navigation pane, click **Services** to view all services.

3.1.4.7.8. Update a service

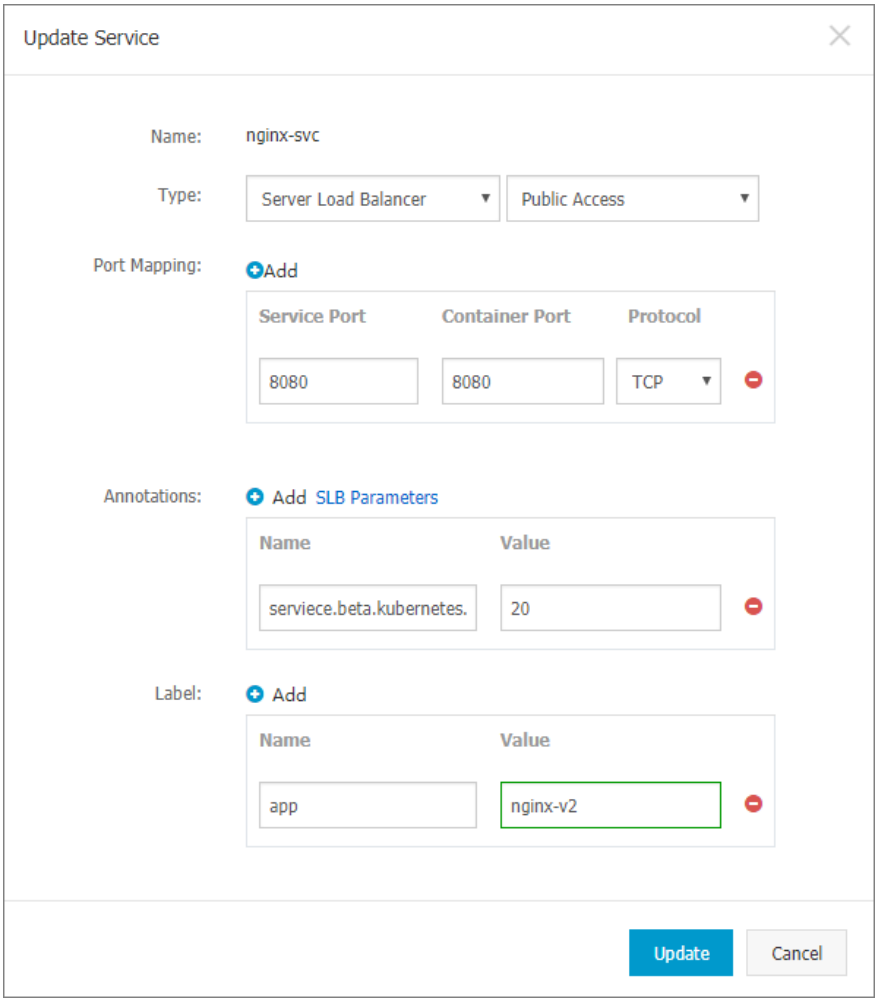
You can update a service through the Container Service console.

Update a service through the Services page

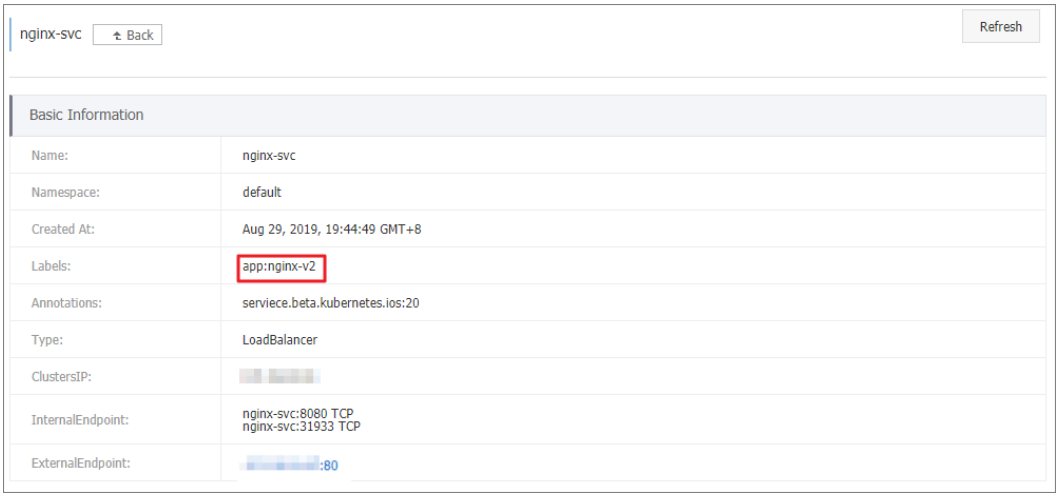
1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Ingresses and Load Balancing > Services**. The Services page appears.
3. Select the target cluster and namespace. Find the target service and click **Update** in the Actions column. In

this example, the target service is nginx-svc.

4. In the dialog box that appears, modify the configurations based on your needs and click **Update**.



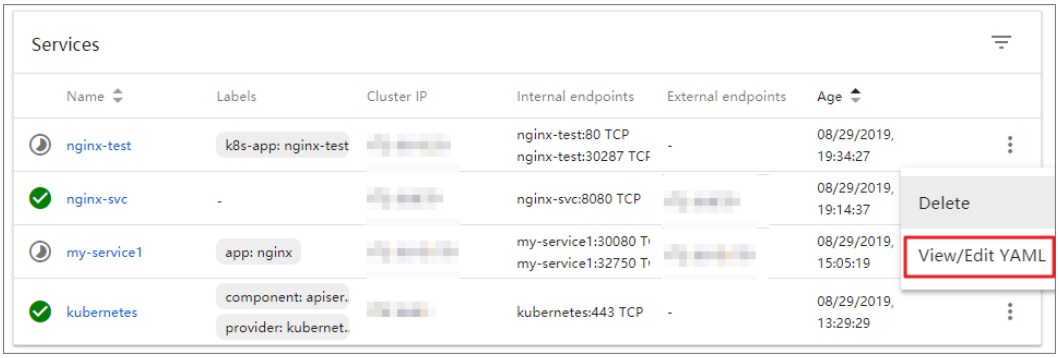
5. Find the service on the Services page and click **Details** in the Actions column to view configuration changes. In this example, the label of the service is updated.



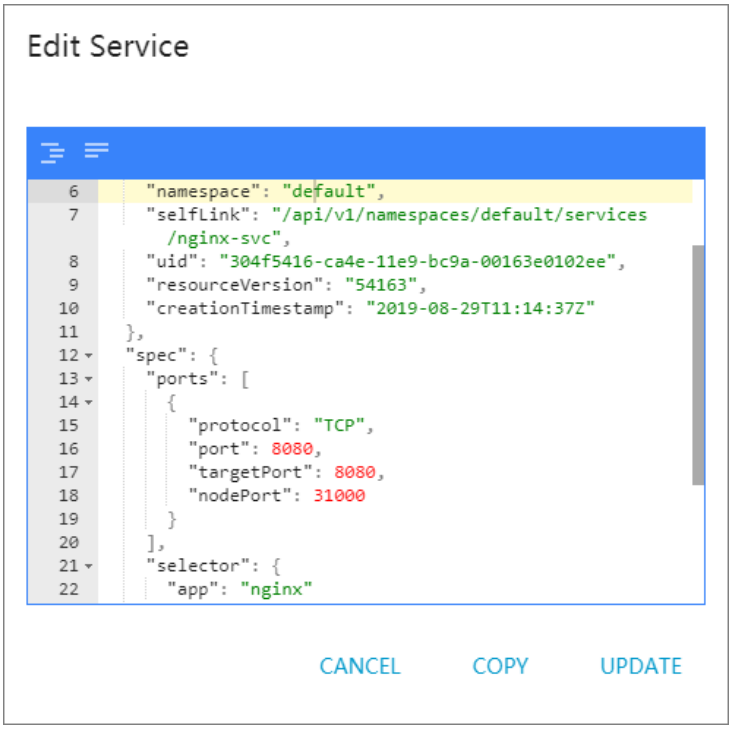
Update a service through Kubernetes Dashboard

1. Log on to the Container Service console.

2. In the left-side navigation pane, choose **Clusters**. The Clusters page appears.
3. Find the target cluster and click **Dashboard** in the Actions column.
4. On Kubernetes Dashboard, select the target namespace and click **Services** in the left-side navigation pane.
5. Find the target service, click the More icon on the right, and choose **View/edit YAML**.



6. In the dialog box that appears, modify the configurations. For example, change nodePort to `31000`. Then click **UPDATE**.



3.1.4.7.9. Delete a service

Prerequisites

- You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.
- You have created a service. For more information, see Create Servcies.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Ingresses and Load Balancing > Services**. The Services page appears.

3. Select the target cluster and namespace. Find the target service and click **Delete** in the Actions column. In this example, the target service is nginx-svc.
4. In the dialog box that appears, click **Confirm**. The target service is deleted and disappears from the Services page.

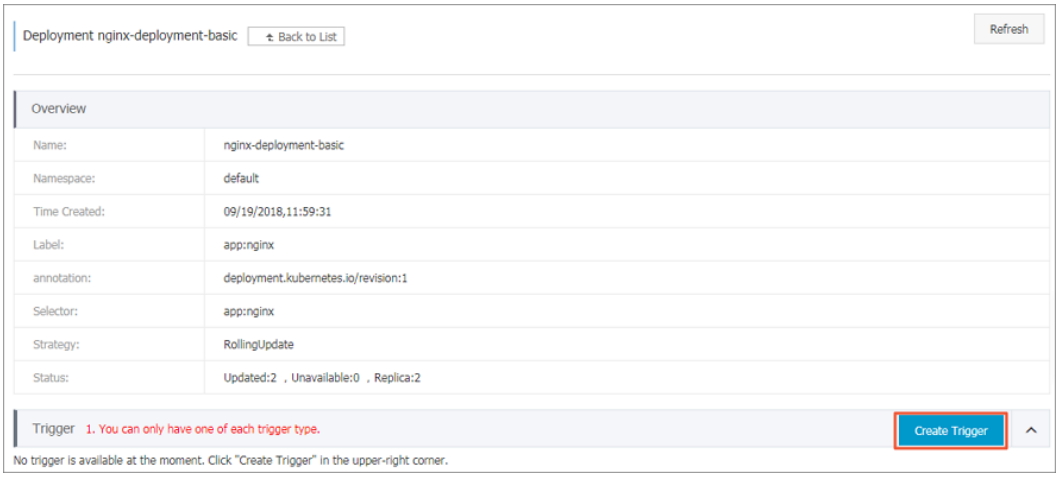
3.1.4.7.10. Create a trigger on an application

Prerequisites

- You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.
- You have created an application based on which the trigger is created and tested. In this example, an NGINX application is created.

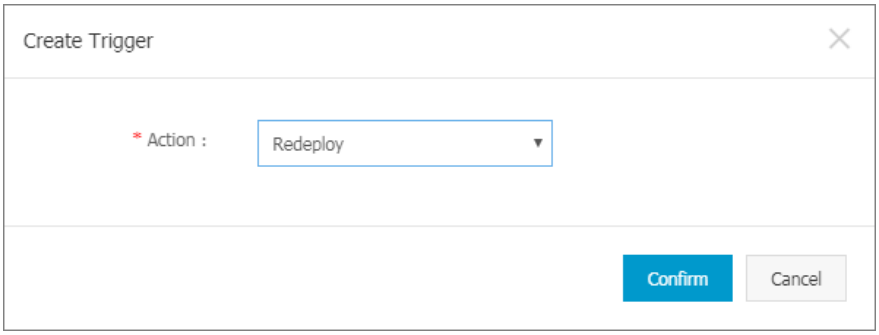
Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Applications > Deployments**. On the **Deployments** page that appears, select the target cluster and namespace. Then find the NGINX application and click **Details** in the Actions column.
3. On the application details page that appears, click **Create Trigger** in the Trigger section.

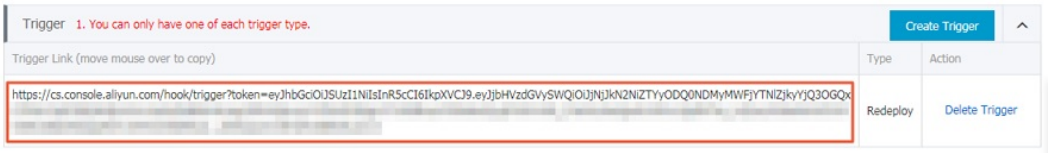


4. In the dialog box that appears, set Action to **Redeploy** and click **OK**.

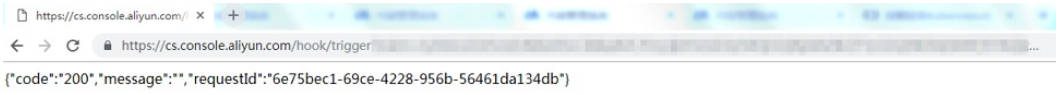
> **Note** Currently, you can only create a trigger to redeploy the application.



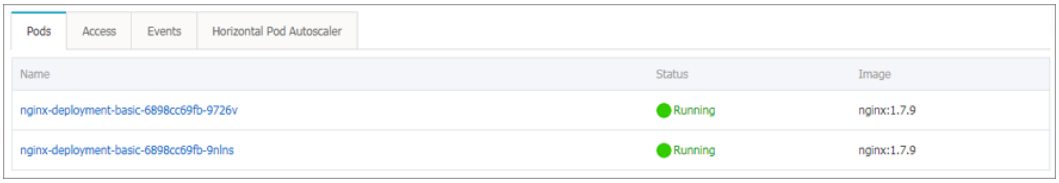
After the trigger is created, a trigger link address is displayed in the Trigger section on the **Deployment nginx** page.



5. Copy the link and open it in your browser. A message appears, displaying information such as the request ID.



6. Go to the **Deployment nginx** page. A new pod is displayed on the **Pods** tab.



After the new pod is successfully deployed, the old pod will be automatically deleted.

What's next

You can call triggers through GET or POST requests from a third-party system. For example, you can run **curl** commands to call triggers.

To call the redeploy trigger, run the following command:

```
curl https://cs.console.aliyun.com/hook/trigger?token=xxxxxxx
```

3.1.4.7.11. View pods

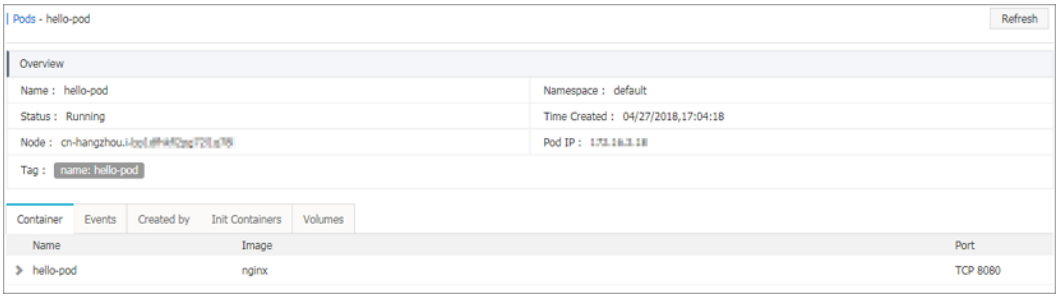
You can view pods through the console.

View pods on the Pods page

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Applications > Pods**. The Pods page appears.
3. Select the target cluster and namespace. Find the target pod and click **View Details**.

> **Note** You can update or delete pods on the Pods page. We recommend that you use deployments to manage pods if they were created by deployments.

4. On the pod details page, you can view detailed information about the pod.



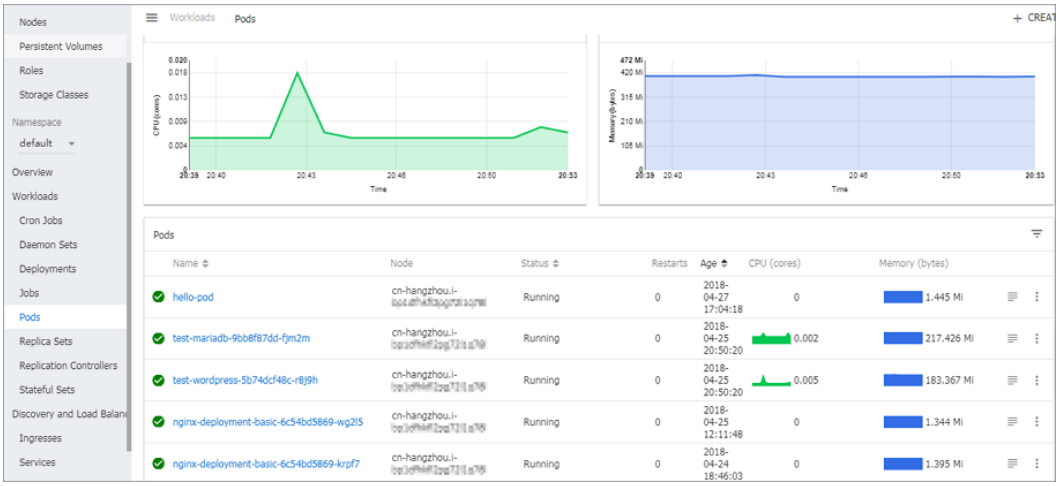
View pods through Kubernetes Dashboard

1. Log on to the Container Service console.

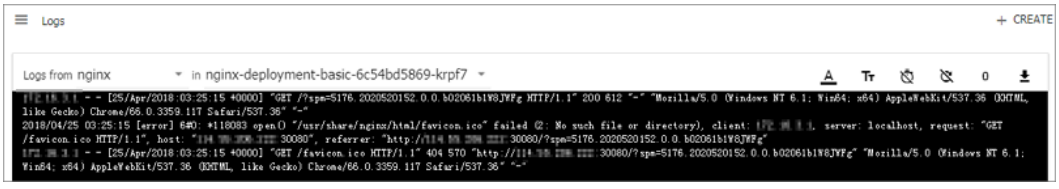
2. In the left-side navigation pane, choose **Clusters > Clusters**. The Clusters page appears.
3. Find the target cluster and click **Dashboard** in the Actions column. The Kubernetes Dashboard page appears.
4. In the left-side navigation pane, click **Pods** to view pods of the cluster.

   You can also click **Services** in the left-side navigation pane and then click a service name to view pods of the service.

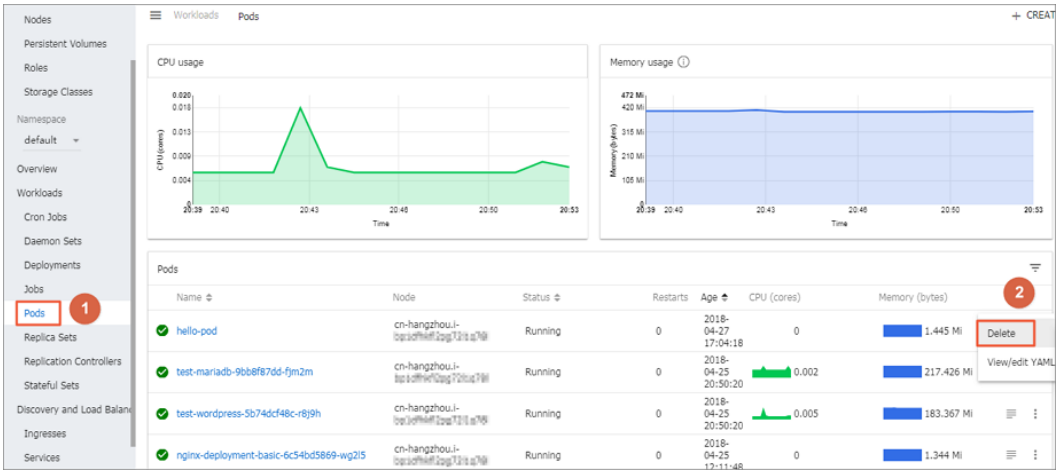
5. The icon at the left side of each pod indicates the status of the pod. indicates that the pod is being deployed. indicates that the pod has been successfully deployed.



6. Select a pod and click its name to view pod details, including the CPU and memory usage.
7. Select a pod and click at the right to view logs.



8. You can also click the More icon and click **Delete** to delete the pod.



3.1.4.7.12. Schedule pods to nodes

You can add labels to nodes and then configure `nodeSelector` to schedule pods to specific nodes. For more information about how nodeSelector works, see nodeselector.

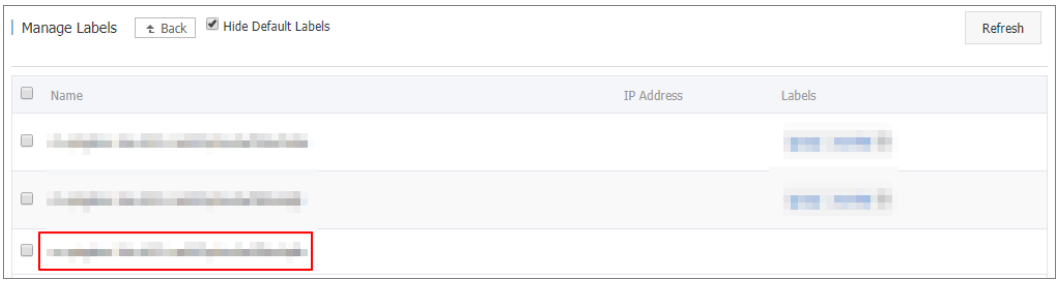
To meet business needs, you may need to deploy a management service on a master node, or deploy certain services on nodes with SSD storage. You can use the following method to schedule pods to specific nodes based on needs.

Prerequisites

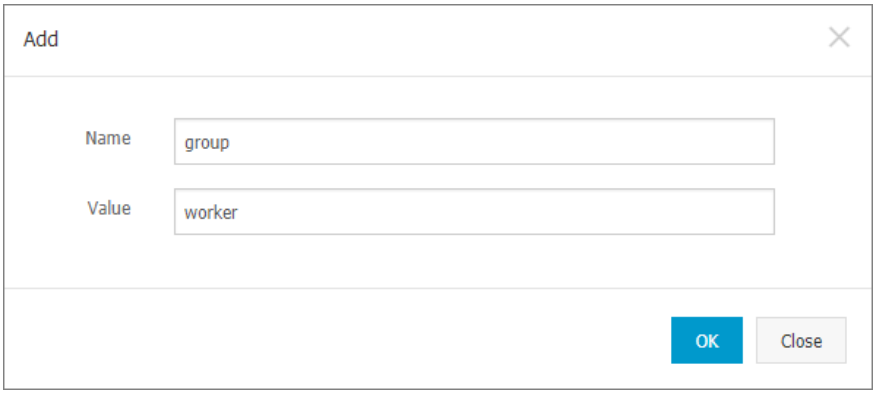
You have created a Kubernetes cluster. For more information, see Create a Kubernetes cluster.

Step 1: Add a label to a node

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Nodes**. The Nodes page appears.
3. Select the target cluster and click **Manage Labels** in the upper-right corner.
4. Select one or more nodes and then click **Add Label**. In this example, select a worker node.



5. In the dialog box that appears, enter the label name and value, and then click **OK**.



On the Manage Labels page, you can find the `group:worker` label next to the selected node.

You can also use the following command to add a label to a node: `kubectl label nodes <node-name> <label-key>=<label-value>`.

Step 2: Schedule a pod to the node

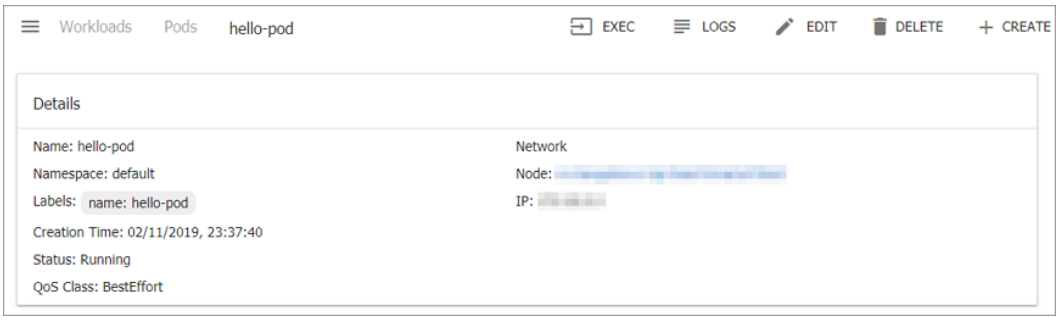
1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Applications > Deployments**. The **Deployments** page appears.
3. In the upper-right corner, click **Create from Template**.
4. Configure the template to create a pod and schedule it to the node from step 1. Then, click **Create**.
   - **Cluster**: Select the cluster where the pod is deployed.
   - **Namespace**: Select the namespace where the pod belongs. In this example, select the default namespace.
   - **Sample Template**: In this example, select Custom.

   Enter the following template content:

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    name: hello-pod
  name: hello-pod
spec:
  containers:
    - image: nginx
      imagePullPolicy: IfNotPresent
      name: hello-pod
      ports:
        - containerPort: 8080
          protocol: TCP
      resources: {}
      securityContext:
        capabilities: {}
        privileged: false
      terminationMessagePath: /dev/termination-log
  dnsPolicy: ClusterFirst
  restartPolicy: Always
  nodeSelector:
    group: worker  ##This value must be the same as the label of the node from step 1.
status: {}
```

5. Click **Create**. A message appears indicating the deployment status. After the pod is created, click

Kubernetes Dashboard to check the status of hello-pod.



3.1.4.7.13. Simplify Kubernetes application deployment by using Helm

This topic introduces the basic concepts and components of Helm and describes how to use Helm to deploy the sample applications WordPress and Spark in a Container Service for Kubernetes cluster.

Prerequisites

- A Kubernetes cluster is created in the Container Service console. For more information, see Create a Kubernetes cluster.

  Tiller is automatically deployed to the cluster when the Kubernetes cluster is created. The Helm command-line interface (CLI) is automatically installed on each master node. You must configure the Helm CLI to point to the Alibaba Cloud chart repository.

- The supported Kubernetes version is used.

  Only Kubernetes 1.8.4 and later versions are supported. For Kubernetes 1.8.1, you can upgrade the cluster to the required version. To upgrade the cluster, log on to the Container Service console, go to the Clusters page, find the cluster, and then choose More > **Upgrade Cluster** in the Actions column for the cluster.

Context

When you run and manage applications with Kubernetes, you can use Helm as the package manager to simplify application distribution and deployment. The Helm project enables consist software packaging and supports version control. In the Container Service console, the App Catalog feature integrates the Helm binaries and supports the Alibaba Cloud chart repository. This allows you to easily deploy applications by using the Helm CLI or in the Container Service console.

Overview

Helm is an open source tool that is created by Deis. It can be used to simplify the deployment and management of Kubernetes applications.

Helm works as a Kubernetes package manager and allows you to discover, share, and run applications that are created in Kubernetes. When you use Helm, you must understand the following basic concepts:

- Chart: a packaging format used by Helm. Each chart contains the images, dependencies, and resource definitions that are required for running an application. A chart may contain service definitions in a Kubernetes cluster. You can use a chart in a similar way as you use a Homebrew formula, the dpkg packages manager of the Advanced Package Tool (APT) package management system, or the Red Hat Package Manager (RPM) package for Yellowdog Updater, Modified (YUM).
- Release: an instance of a chart that runs in a Kubernetes cluster. A chart can be installed many times into the same cluster. After a chart is installed, a new release is created. For example, you can install a MySQL chart. If you want to run two databases in your cluster, you can install the MySQL chart twice. Each installation generates a release with a release name.

- Repository: the location where charts are stored and released.

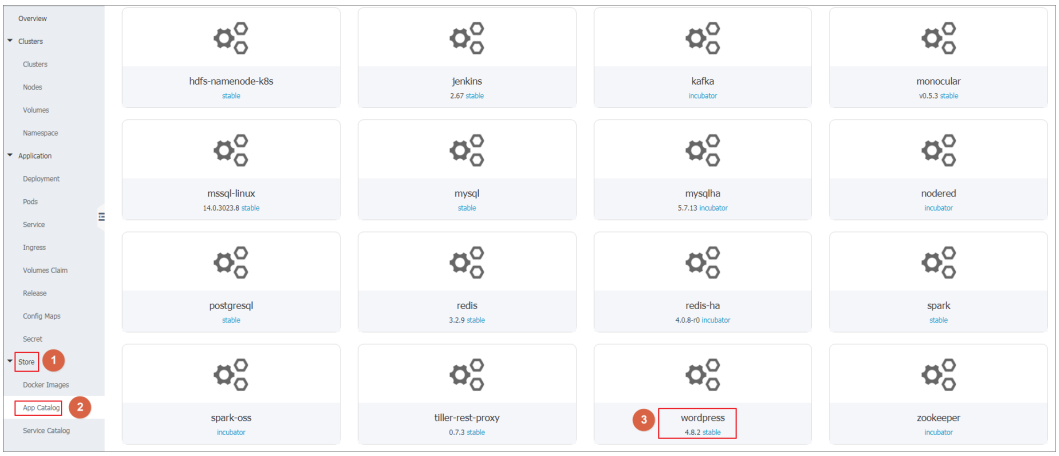
Helm components

Helm works in a client-server architecture and consists of the following components:

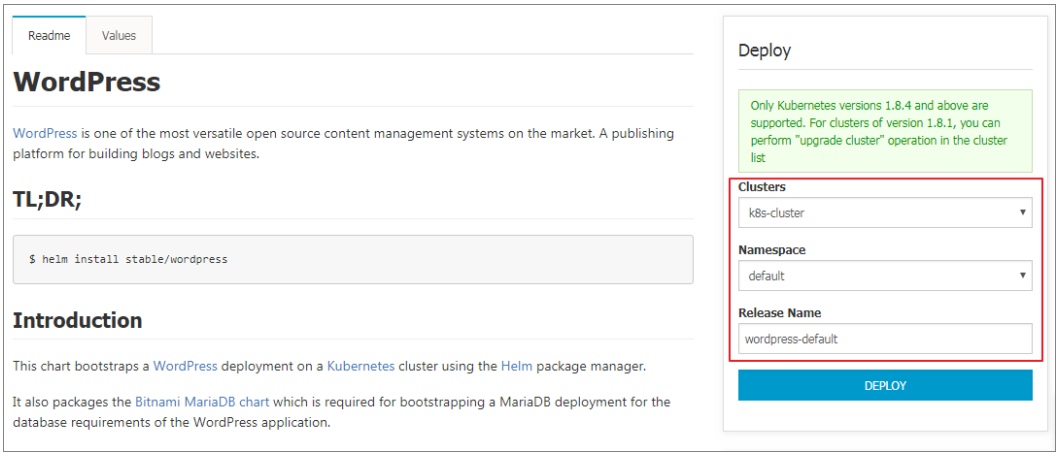
- The Helm CLI is the Helm client that runs on your on-premises computer or on the master nodes of a Kubernetes cluster.
- Tiller is the server-side component and runs on a Kubernetes cluster. Tiller manages the lifecycles of Kubernetes applications.
- A repository is used to store charts. The Helm client can access the index file and packaged charts in a chart repository over HTTP.

Deploy an application in the Container Service console

1. Log on to the Container Service console
2. In the left-side navigation pane, choose **Marketplace > App Catalog** to go to the App Catalog page.
3. Click a chart, for example, WordPress, to go to the page that shows the details of the chart.



4. In the Deploy section on the right of the page, enter the basic information for the deployment.
   - **Cluster**: Select the cluster to which you want to deploy the application.
   - **Namespace**: Select a namespace for the application. By default, this parameter is set to default.
   - **Release Name**: Enter a release name for the application.

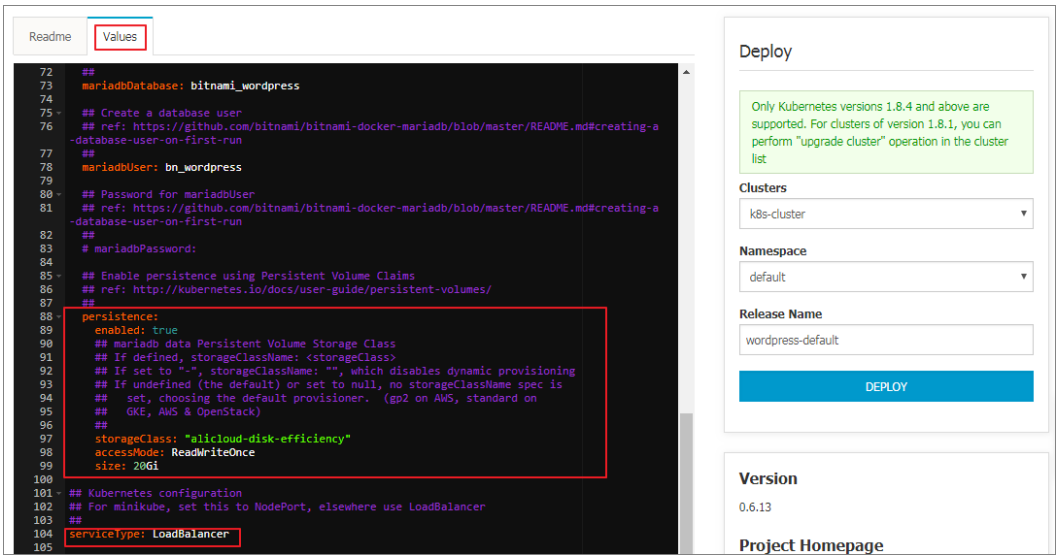


5. Click the **Parameters** tab to set the parameters.

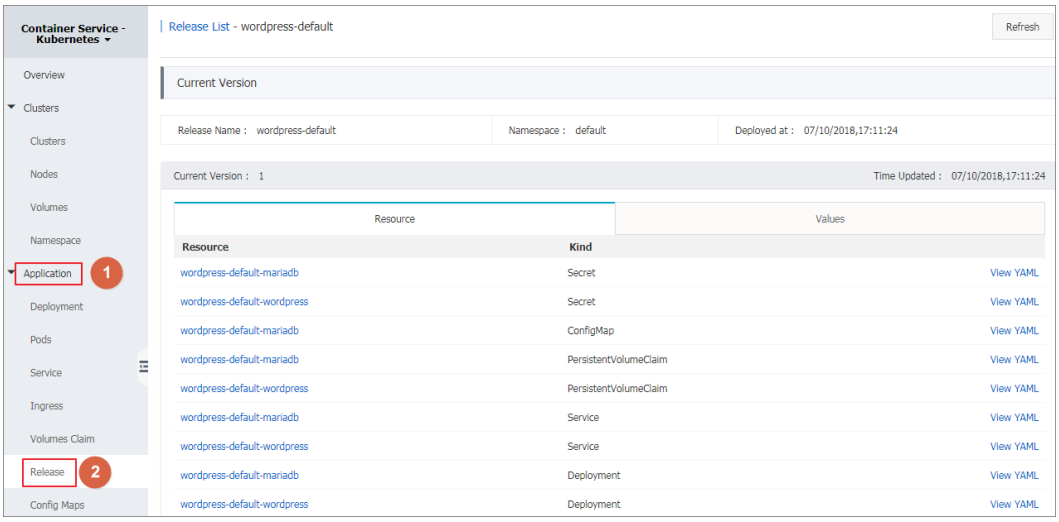
   In this example, a dynamically provisioned volume is associated with a persistent volume claim (PVC). For more

information, see Use Apsara Stack disks.

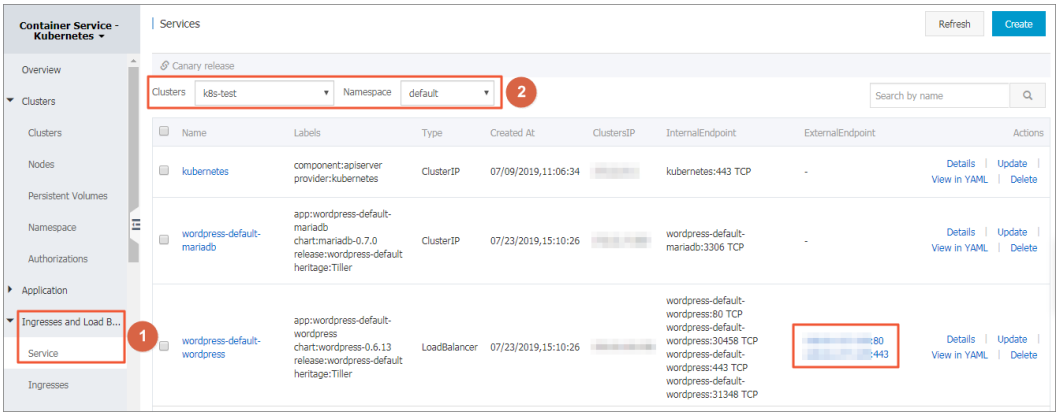
> **Note** Before the association, you must create a persistent volume (PV) as the dynamic volume. The capacity of the PV cannot be less than the value defined by the PVC.



6. After you set the parameters, click **Create** to deploy the application. After the application is deployed, you are navigated to the release page of the application.



7. In the left-side navigation pane, choose **Ingresses and Load Balancing > Services**, select the target cluster and namespace, and then find the required service from the list of services. In the External Endpoint column for the service, you can see the external HTTP and HTTPS endpoints.



8. Click either of the endpoints to go to the WordPress application where you can publish blog posts.

Deploy an application by using the Helm CLI

After the Helm CLI is automatically installed on the master node of the Kubernetes cluster and points to the required repository, you can log on to the master node by using SSH. This allows you to deploy applications by using the Helm CLI. For more information, see Connect to a master node through SSH. You can also install and configure the Helm CLI and kubectl on your on-premises computer.

In this example, on your on-premises computer, the Helm CLI and kubectl are installed and configured and the WordPress and Spark applications are deployed.

1. Install and configure the Helm CLI and kubectl.
   i. Install and configure kubectl on your on-premises computer.

      For more information, see Connect to a Kubernetes cluster through kubectl.

      To view the information of the target Kubernetes cluster, on the command line, enter `kubectl cluster-info`.

   ii. Install Helm on your on-premises computer.

      For more information, see Install Helm.
2. Deploy the WordPress application.

   To deploy a WordPress blog website by using Helm, perform the following steps:

   i. On the command line, run the following command:

   ```
   helm install --name wordpress-test stable/wordpress
   ```

   > **Note** Container Service for Kubernetes allows you to use block storage or disks as dynamically provisioned volumes. Before you deploy the WordPress application, you must create dynamically provisioned volumes based on disks.

   The following example shows the output:

   ```
   NAME:   wordpress-test
   LAST DEPLOYED: Mon Nov  20 19:01:55 2017
   NAMESPACE: default
   STATUS: DEPLOYED
   ...
   ```

   ii. On the command line, run the following commands to view the release and service of WordPress.

   ```
   helm list
   kubectl get svc
   ```

iii. On the command line, run the following command to view the pod that is associated with the WordPress application. The pod may take a few minutes to change to the running state.

```
kubectl get pod
```

iv. On the command line, run the following command to obtain the endpoint of the WordPress application.

```
echo http://$(kubectl get svc wordpress-test-wordpress -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
```

You can enter the preceding endpoint in your browser to access the WordPress application.

You can also follow the chart instructions described in the console and run the following commands to obtain the administrator account and password of the WordPress application:

```
echo Username: user
echo Password: $(kubectl get secret --namespace default wordpress-test-wordpress -o jsonpath="{.data.wordpress-password}" | base64 --decode)
```

v. On the command line, run the following command to delete the WordPress application:

```
helm delete --purge wordpress-test
```

Use a third-party chart repository

You can use the default Alibaba Cloud chart repository. You can also use a third-party chart repository if the third-party chart repository is accessible. On the command line, run the following command to add a third-party chart repository:

```
helm repo add Repository name Repository URL
helm repo update
```

For more information about Helm commands, see Helm documentation.

References

The Kubernetes community has experienced rapid technological developments based on Helm. These developments have allowed software providers, such as Bitnami, to offer high-quality charts. For more information about available charts, visit `https://kubeapps.com/` .

3.1.4.8. SLB and Ingress

3.1.4.8.1. Overview

Container Service allows you to flexibly manage load balancing and customize load balancing policies for Kubernetes clusters. Kubernetes clusters provide you with a variety of methods to access containerized applications. They also allow you to use SLB or Ingress to access internal services and implement load balancing.

3.1.4.8.2. Use SLB to access Services

You can access a Service through Server Load Balancer (SLB).

Use the CLI

1. Create an NGINX application by using the command-line interface (CLI).

```
root@master # kubectl run nginx --image=registry.aliyuncs.com/acs/netdia:latest
       root@master # kubectl get po
       NAME                                READY     STATUS    RESTARTS   AGE
       nginx-2721357637-dvwq3              1/1       Running   1          6s
```

2. Create a Service for the NGINX application and set `type=LoadBalancer` to expose the NGINX Service to the Internet through an SLB instance.

```
root@master # kubectl expose deployment nginx --port=80 --target-port=80 --type=LoadBalancer
       root@master # kubectl get svc
       NAME                  CLUSTER-IP      EXTERNAL-IP      PORT(S)                       AGE
       nginx                 172.19.XX.XX    101.37.XX.XX     80:31891/TCP                  4s
```

3. Copy `http://101.37.XX.XX` into the address bar of your browser and press Enter to access the NGINX Service.

SLB parameters

SLB provides a variety of parameters that you can use to configure features and services such as health check, billing method, and SLB instance type. For more information, see SLB parameters.

Annotations

You can add annotations to use the load balancing features provided by SLB.

Use an existing internal-facing SLB instance

You need to add two annotations. You must replace "yourloadbalancer-id" with your SLB instance ID.

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    service.beta.kubernetes.io/alicloud-loadbalancer-address-type: intranet
    service.beta.kubernetes.io/alicloud-loadbalancer-id: your-loadbalancer-id
  labels:
    run: nginx
  name: nginx
  namespace: default
spec:
  ports:
  - name: web
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    run: nginx
  sessionAffinity: None
  type: LoadBalancer
```

Save the preceding code as an slb.svc file and run the following command: `kubectl apply -f slb.svc`.

Create an HTTPS-based Service of the Loadbalancer type

You must first create a certificate in the SLB console. Then, you can use the certificate ID (cert-id) and the following template to create a LoadBalancer Service and an HTTPS-based SLB instance.

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    service.beta.kubernetes.io/alicloud-loadbalancer-cert-id: your-cert-id
    service.beta.kubernetes.io/alicloud-loadbalancer-protocol-port: "https:443"
  labels:
    run: nginx
  name: nginx
  namespace: default
spec:
  ports:
  - name: web
    port: 443
    protocol: TCP
    targetPort: 443
  selector:
    run: nginx
  sessionAffinity: None
  type: LoadBalancer
```

Note Annotations are case sensitive.

SLB parameters

Annotation	Description	Default value
service.beta.kubernetes.io/alicloud-loadbalancer-protocol-port	The listening port. Separate multiple ports with commas (,). Example: https:443,http:80.	N/A
service.beta.kubernetes.io/alicloud-loadbalancer-address-type	The type of the SLB instance. Valid values: internet and intranet.	internet
service.beta.kubernetes.io/alicloud-loadbalancer-slb-network-type	The network type of the SLB instance. Valid values: classic and vpc.	classic
service.beta.kubernetes.io/alicloud-loadbalancer-charge-type	The billing method of the SLB instance. Valid values: paybytraffic and paybybandwidth.	paybybandwidth
service.beta.kubernetes.io/alicloud-loadbalancer-id	The ID of the SLB instance. You can set the loadbalancer-id parameter to specify an existing SLB instance and its existing listeners will be overwritten. The SLB instance will not be deleted if the Service is deleted.	N/A
service.beta.kubernetes.io/alicloud-loadbalancer-backend-label	The labels that specify the nodes to be added as backend servers of the SLB instance.	N/A
service.beta.kubernetes.io/alicloud-loadbalancer-region	The region where the SLB instance is deployed.	N/A
service.beta.kubernetes.io/alicloud-loadbalancer-bandwidth	The bandwidth of the SLB instance.	50
service.beta.kubernetes.io/alicloud-loadbalancer-cert-id	The certificate ID. You must upload the certificate first.	""

Annotation	Description	Default value
service.beta.kubernetes.io/alicloud-loadbalancer-health-check-flag	Valid values: on and off.	Default value: off. If TCP is used, do not modify this parameter. The health check feature is enabled for TCP listeners by default and cannot be disabled.
service.beta.kubernetes.io/alicloud-loadbalancer-health-check-type	For more information, see *CreateLoadBalancerTCPListener* in the *SLB Developers Guide*.	N/A
service.beta.kubernetes.io/alicloud-loadbalancer-health-check-uri	For more information, see *CreateLoadBalancerTCPListener* in the *SLB Developers Guide*.	N/A
service.beta.kubernetes.io/alicloud-loadbalancer-health-check-connect-port	For more information, see *CreateLoadBalancerTCPListener* in the *SLB Developers Guide*.	N/A
service.beta.kubernetes.io/alicloud-loadbalancer-healthy-threshold	For more information, see *CreateLoadBalancerTCPListener* in the *SLB Developers Guide*.	N/A
service.beta.kubernetes.io/alicloud-loadbalancer-unhealthy-threshold	For more information, see *CreateLoadBalancerTCPListener* in the *SLB Developers Guide*.	N/A
service.beta.kubernetes.io/alicloud-loadbalancer-health-check-interval	For more information, see *CreateLoadBalancerTCPListener* in the *SLB Developers Guide*.	N/A
service.beta.kubernetes.io/alicloud-loadbalancer-health-check-connect-timeout	For more information, see *CreateLoadBalancerTCPListener* in the *SLB Developers Guide*.	N/A
service.beta.kubernetes.io/alicloud-loadbalancer-health-check-timeout	For more information, see *CreateLoadBalancerTCPListener* in the *SLB Developers Guide*.	N/A

3.1.4.8.3. Configure ingress monitoring

You can enable the default VTS module to view ingress monitoring data.

Enable VTS module by using the CLI

1. Modify the ingress ConfigMap to add the following configuration item: `enable-vts-status: "true"` .

```
root@master # kubectl edit configmap nginx-configuration -n kube-system
      configmap "nginx-configuration" edited
```

The modified ingress ConfigMap is as follows:

```
apiVersion: v1
data:
  enable-vts-status: "true" # Enable the VTS module
  proxy-body-size: 20m
kind: ConfigMap
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","data":{"proxy-body-size":"20m"},"kind":"ConfigMap","metadata":{"annotat
ions":{},"labels":{"app":"ingress-nginx"},"name":"nginx-configuration","namespace":"kube-system"}
}
  creationTimestamp: 2018-03-20T07:10:18Z
  labels:
    app: ingress-nginx
  name: nginx-configuration
  namespace: kube-system
  selfLink: /api/v1/namespaces/kube-system/configmaps/nginx-configuration
```

2. Verify that the VTS module is enabled.

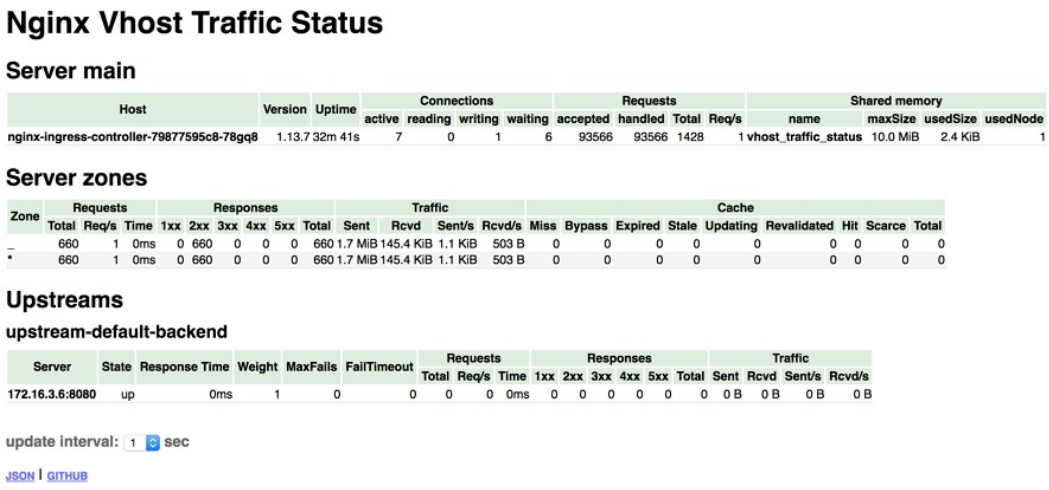
```
root@master # kubectl get pods --selector=app=ingress-nginx -n kube-system
NAME                                        READY     STATUS    RESTARTS   AGE
nginx-ingress-controller-79877595c8-78gq8   1/1       Running   0          1h
root@master # kubectl exec -it nginx-ingress-controller-79877595c8-78gq8 -n kube-system -- cat /e
tc/nginx/nginx.conf | grep vhost_traffic_status_display
        vhost_traffic_status_display;
        vhost_traffic_status_display_format html;
```

3. Access the NGINX Ingress Controller from a local computer.

> **Note** By default, the VTS port is not enabled for security concerns. The following example uses port forwarding to access the controller.

```
root@master # kubectl port-forward nginx-ingress-controller-79877595c8-78gq8 -n kube-system 18080
Forwarding from 127.0.0.1:18080 -> 18080
Handling connection for 18080
```

4. Visit `http://localhost:18080/nginx_status` to access the NGINX Ingress Controller.



Enable VTS module by using the Kubernetes Dashboard

1. Log on to the Container Service console.

2. Find the target cluster and click **Dashboard** in the Actions column to go to Kubernetes Dashboard.
3. In the left-side navigation pane, select the kube-system namespace and edit the nginx-configuration ConfigMap to add the following configuration item: `enable-vts-status: "true"` .

   The modified ingress ConfigMap is as follows:

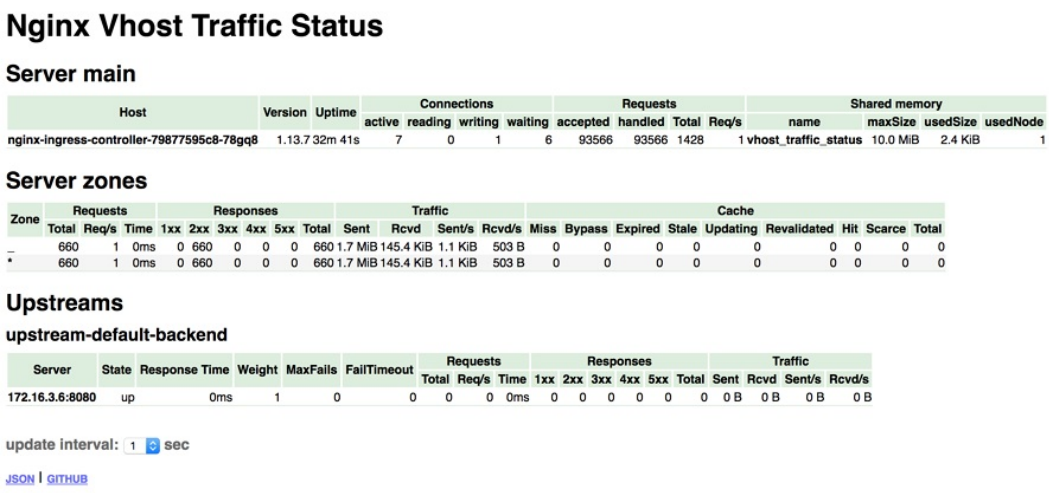
```
{
  "kind": "ConfigMap",
  "apiVersion": "v1",
  "metadata": {
    "name": "nginx-configuration",
    "namespace": "kube-system",
    "selfLink": "/api/v1/namespaces/kube-system/configmaps/nginx-configuration",
    "creationTimestamp": "2018-03-20T07:10:18Z",
    "labels": {
      "app": "ingress-nginx"
    },
    "annotations": {
      "kubectl.kubernetes.io/last-applied-configuration": "{\"apiVersion\":\"v1\",\"data\":{\"pro
xy-body-size\":\"20m\"},\"kind\":\"ConfigMap\",\"metadata\":{\"annotations\":{},\"labels\":{\"app
\":\"ingress-nginx\"},\"name\":\"nginx-configuration\",\"namespace\":\"kube-system\"}}\n"
    }
 },
 "data": {
   "proxy-body-size": "20m",
   "enable-vts-status": "true"
 }
}
```

4. Access the NGINX Ingress Controller from a local computer.

   > **Note** By default, the VTS port is not enabled for security concerns. The following example uses port forwarding to access the controller.

```
root@master # kubectl port-forward nginx-ingress-controller-79877595c8-78gq8 -n kube-system 18080
Forwarding from 127.0.0.1:18080 -> 18080
Handling connection for 18080
```

5. Visit `http://localhost:18080/nginx_status` to access the NGINX Ingress Controller.



Nginx Vhost Traffic Status

Server main

Host	Version	Uptime	Connections				Requests				Shared memory			
			active	reading	writing	waiting	accepted	handled	Total	Req/s	name	maxSize	usedSize	usedNode
nginx-ingress-controller-79877595c8-78gq8	1.13.7	32m 41s	7	0	1	6	93566	93566	1428	1	vhost_traffic_status	10.0 MiB	2.4 KiB	1

Server zones

Zone	Requests			Responses					Traffic				Cache								
	Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd	Sent/s	Rcvd/s	Miss	Bypass	Expired	Stale	Updating	Revalidated	Hit	Scarce	Total
_	660	1	0ms	0	660	0	0	0	660	1.7 MiB	145.4 KiB	1.1 KiB	503 B	0	0	0	0	0	0	0	0	0
*	660	1	0ms	0	660	0	0	0	660	1.7 MiB	145.4 KiB	1.1 KiB	503 B	0	0	0	0	0	0	0	0	0

Upstreams

upstream-default-backend

Server	State	Response Time	Weight	MaxFails	FailTimeout	Requests			Responses						Traffic			
						Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd	Sent/s	Rcvd/s
172.16.3.6:8080	up	0ms	1	0	0	0	0	0ms	0	0	0	0	0	0	0 B	0 B	0 B	0 B

update interval: 1 sec

JSON | GITHUB

3.1.4.8.4. Ingress support

Kubernetes clusters support ingress rules. You can define ingress rules based on your needs to implement load balancing.

In Kubernetes, an ingress is a collection of routing rules that authorize external access to cluster services. You can use an ingress to enable Layer-7 load balancing. You can configure an ingress to provide Kubernetes services with externally reachable URLs, SLB instances, SSL connections, and name-based virtual hosting.

Prerequisites

To test a complex routing scenario, this example creates an NGINX application that consists of multiple services. You need to create an NGINX deployment and multiple services in advance. In practice, replace service names with the actual values.

```
root@master # kubectl run nginx --image=registry.cn-hangzhou.aliyuncs.com/acs/netdia:latest
    root@master # kubectl expose deploy nginx --name=http-svc --port=80 --target-port=80
    root@master # kubectl expose deploy nginx --name=http-svc1 --port=80 --target-port=80
    root@master # kubectl expose deploy nginx --name=http-svc2 --port=80 --target-port80
    root@master # kubectl expose deploy nginx --name=http-svc3 --port=80 --target-port=80
```

Simple routing

Use the following commands to create a simple ingress that redirects traffic to path */svc* to a http-svc service. `nginx.ingress.kubernetes.io/rewrite-target: /` indicates that traffic to */svc* is redirected to the root path */*, which can be recognized by the backend service.

```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: simple
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - http:
      paths:
      - path: /svc
        backend:
          serviceName: http-svc
          servicePort: 80
EOF
```

```
root@master # kubectl get ing
NAME            HOSTS         ADDRESS          PORTS     AGE
simple          *             101.37.192.211   80        11s
```

You can now visit `http://101.37.192.211/svc` to access the NGINX service.

Simple fanout based on domains

If you have multiple services exposed externally through different domains, you can use the following configuration to implement a simple fanout based on domains:

```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: simple-fanout
spec:
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /foo
        backend:
          serviceName: http-svc1
          servicePort: 80
      - path: /bar
        backend:
          serviceName: http-svc2
          servicePort: 80
  - host: foo.example.com
    http:
      paths:
      - path: /film
        backend:
          serviceName: http-svc3
          servicePort: 80
EOF
```

```
root@master # kubectl get ing
NAME            HOSTS         ADDRESS          PORTS     AGE
simple-fanout   *             101.37.192.211   80        11s
```

You can now visit `http://foo.bar.com/foo` to access service `http-svc1`, visit `http://foo.bar.com/bar` to access service `http-svc2`, and visit `http://foo.example.com/film` to access service `http-svc3`.

Note

- In a production environment, you need to point the domain to the returned address `101.37.192.211`.
- In a test environment, you need to add the following mapping rules to the `hosts` file.

```
101.37.192.211 foo.bar.com
101.37.192.211 foo.example.com
```

Default domain for simple routing

If you have no domain address, you can use the default domain associated with the ingress to access the service. The default domain is in the following format: `*.[cluster-id].[region-id].alicontainer.com`. You can obtain the domain address on the cluster basic information page in the console.

You can use the following configuration to expose the two services through the default domain.

```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: shared-dns
spec:
  rules:
  - host: foo.[cluster-id].[region-id].alicontainer.com  ## Replace with the default domain of your c
luster
    http:
      paths:
      - path: /
        backend:
          serviceName: http-svc1
          servicePort: 80
  - host: bar.[cluster-id].[region-id].alicontainer.com  ## Replace with the default domain of your c
luster
    http:
      paths:
      - path: /
        backend:
          serviceName: http-svc2
          servicePort: 80
EOF
```

```
root@master # kubectl get ing
NAME            HOSTS         ADDRESS          PORTS     AGE
shared-dns   foo.[cluster-id].[region-id].alicontainer.com,bar.[cluster-id].[region-id].alicontainer.
com             47.95.160.171   80        40m
```

You can now visit `http://foo.[cluster-id].[region-id].alicontainer.com/` to access service `http-svc1` and visit `http://bar.[cluster-id].[region-id].alicontainer.com` to access service `http-svc2`.

Secure routing

Container Service supports managing multiple certificates to enhance protection for your services.

1. Prepare your certificate.

   If you have no certificate, use the following method to generate a test certificate.

   > **Note** The domain must be the same as the one specified in your ingress configuration.

   ```
   root@master # openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out tls.crt -s
   ubj "/CN=foo.bar.com/O=foo.bar.com"
   ```

   After you run the preceding command, a certificate file *tls.crt* and a private key file *tls.key* are generated.

   Use the certificate and private key to create a Kubernetes secret named *foo.bar*. You need to reference the secret when you create the ingress.

   ```
   root@master # kubectl create secret tls foo.bar --key tls.key --cert tls.crt
   ```

2. Create a secure ingress.

```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: tls-fanout
spec:
  tls:
  - hosts:
    - foo.bar.com
    secretName: foo.bar
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /foo
        backend:
          serviceName: http-svc1
          servicePort: 80
      - path: /bar
        backend:
          serviceName: http-svc2
          servicePort: 80
EOF
```

```
root@master # kubectl get ing
NAME            HOSTS         ADDRESS          PORTS     AGE
tls-fanout      *             101.37.192.211   80        11s
```

3. As described in **Simple fanout based on domains**, you need to configure the `hosts` file or set a domain to access the tls ingress.

   You can visit `http://foo.bar.com/foo` to access service `http-svc1` and visit `http://foo.bar.com/bar` to access service `http-svc2`.

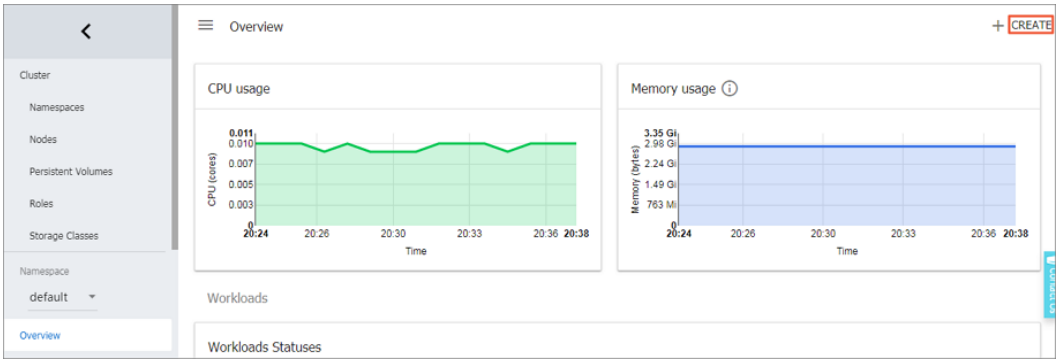
   You can also access the HTTPS service by using HTTP. By default, the ingress redirects HTTP traffic to the HTTPS address. Traffic to `http://foo.bar.com/foo` is automatically redirected to `https://foo.bar.com/foo`.

Deploy an ingress in Kubernetes Dashboard

1. Create an `nginx-svc.yml` file with the following code:

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: simple
spec:
  rules:
  - http:
      paths:
      - path: /svc
        backend:
          serviceName: http-svc
          servicePort: 80
```

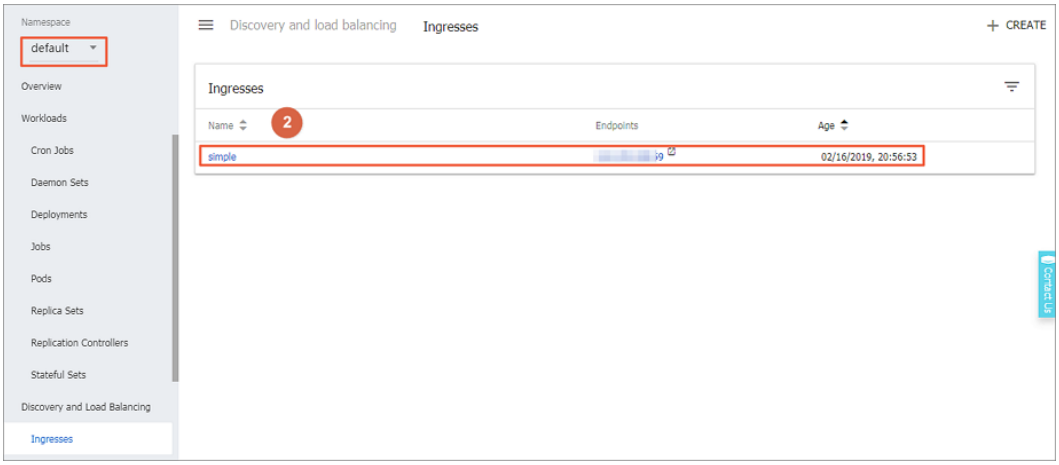
2. Log on to the Container Service console.
3. In the left-side navigation pane, click Clusters. Find the target cluster and click **Dashboard** in the Actions column to go to the **Overview** page.
4. Click **CREATE** in the upper-right corner to create an ingress.



5. Click the **CREATE FROM FILE** tab. Select file `nginx-ingress.yml`.
6. Click **UPLOAD**.

   This creates an ingress that routes Layer-7 traffic for service `http-svc`.
7. In the left-side navigation pane, select the **default** namespace and click **Ingresses**.

   You can view the newly created ingress and its endpoint.



8. Enter the endpoint into your browser to access the `http-svc` service.

3.1.4.8.5. Ingress configurations

Container Service provides Ingress controller components. Integrated with Apsara Server Load Balancer, these components provide Kubernetes clusters with flexible and reliable Ingress service.

An Ingress orchestration template is provided below. When you configure an Ingress through the console, you need to configure annotations and may need to create dependencies. For more information, see Create an ingress through the console, Ingress support, and Kubernetes Ingress. You can also create ConfigMaps to configure Ingresses. For more information, see https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/configmap/.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/service-match: 'new-nginx: header("foo", /^bar$/)' #Canary release ru
le. In this example, the request header is used.
    Nginx. ingress. kubernetes. IO/service-weight: 'New-nginx: 50, old-nginx: 50' #The route weight.
  creationTimestamp: null
  generation: 1
  name: nginx-ingress
  selfLink: /apis/extensions/v1beta1/namespaces/default/ingresses/nginx-ingress
spec:
  rules: ##The Ingress rule.
  - host: foo.bar.com
    http:
      paths:
      - backend:
          serviceName: new-nginx
          servicePort: 80
        path: /
      - backend:
          serviceName: old-nginx
          servicePort: 80
        path: /
tls:            ## Enable TLS for secure routing.
  - hosts:
    - *.xxxxxx.cn-hangzhou.alicontainer.com
    - foo.bar.com
    secretName: nginx-ingress-secret     ##The Secret name.
status:
  loadBalancer: {}
```

Annotations

For each Ingress, you can configure its annotations, Ingress controller, and rules, such as the route weight, canary release rule, and rewrite rules. For more information about annotations, see https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/annotations/.

For example, the following rewrite annotation, `nginx.ingress.kubernetes.io/rewrite-target: /`, indicates that */path* is redirected to the root path */*, which can be recognized by the backend service.

Rules

Ingress rules are used to manage external access to the services in the cluster and can be HTTP or HTTPS rules. You can configure the following items in rules: domain name (virtual hostname), URL path, service name, and port.

For each rule, you need to set the following parameters:

- Domain: The test domain or virtual hostname of your service, such as `foo.bar.com`.
- Path: The URL path of your service. Each path is associated with a backend service. Server Load Balancer only forwards traffic to the backend if the incoming request matches the domain and path.
- Service: Specify the service in the form of `service:port`. You also need to specify a route weight for each service. The Ingress routes traffic to the matching service based on the route weight.
  - Name: The name of the backend service.
  - Port: The port of the service.

- Weight: The route weight of the service in the service group.

  > **Note**
  >
  > a. The weight is a percentage value. For example, you can set two services to the same weight of 50%.
  >
  > b. A service group includes services that have the same domain and path defined in the Ingress configuration. If no weight is set for a service, the default value, 100, is used.

Canary release

Container Service supports multiple traffic splitting approaches to suit scenarios such as canary release and A/B testing.

> **Note** Currently, only Ingress controllers of 0.12.0-5 and later versions support traffic splitting.

1. Traffic splitting based on request header
2. Traffic splitting based on cookie
3. Traffic splitting based on query parameter

After canary release is configured, only requests that match certain rules are routed to the corresponding service. If the weight of the corresponding service is lower than 100%, requests that match certain rules are routed to one of the services in the service group based on the weight.

TLS

You can use a Secret that contains a TLS private key and certificate to encrypt the Ingress. This ensures secure routing. The TLS Secret must contain a certificate named tls.crt and a private key named tls.key. For more information about how TLS works, see TLS. For how to create a Secret, see Configure a secure Ingress.

Labels

You can add labels to the Ingress.

3.1.4.8.6. Create an Ingress in the console

The Container Service console is integrated with the Ingress service. You can create an Ingress in the console and manage inbound traffic that is forwarded to different services to meet your business requirements.

Prerequisites

- A Kubernetes cluster is created and an Ingress controller runs normally in the cluster. For more information, see Create a Kubernetes cluster.
- You can use kubectl to connect to a master node. For more information, see Connect to a Kubernetes cluster through kubectl.
- In this example, the image is retrieved over the Internet. You can replace the image address with the one that is accessible to your own cluster. Otherwise, you can build the image that is used in this example and push the image to a repository. To use the image, you can pull the image from the repository.

Step 1: Create a deployment and a service

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Applications > Deployments**. The **Deployments** page appears.
3. Click **Create from Template** in the upper-right corner of the page.
4. Select the target cluster and namespace, select a sample template or enter a custom template, and then click **Create**.

In this example, two NGINX applications are created. One is named old-nginx and the other is named new-nginx.

The following code example shows the template that is used to create old-nginx:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: old-nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      run: old-nginx
  template:
    metadata:
      labels:
        run: old-nginx
    spec:
      containers:
      - image: registry.cn-hangzhou.aliyuncs.com/xianlu/old-nginx
        imagePullPolicy: Always
        name: old-nginx
        ports:
        - containerPort: 80
          protocol: TCP
      restartPolicy: Always
---
apiVersion: v1
kind: Service
metadata:
  name: old-nginx
spec:
  ports:
  - port: 80
    protocol: TCP
    targetPort: 80
  selector:
    run: old-nginx
  sessionAffinity: None
  type: NodePort
```

The following code example shows the template that is used to create new-nginx:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: new-nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      run: new-nginx
  template:
    metadata:
      labels:
        run: new-nginx
    spec:
      containers:
      - image: registry.cn-hangzhou.aliyuncs.com/xianlu/new-nginx
        imagePullPolicy: Always
        name: new-nginx
        ports:
        - containerPort: 80
          protocol: TCP
      restartPolicy: Always
---
apiVersion: v1
kind: Service
metadata:
  name: new-nginx
spec:
  ports:
  - port: 80
    protocol: TCP
    targetPort: 80
  selector:
    run: new-nginx
  sessionAffinity: None
  type: NodePort
```

5. In the left-side navigation pane, choose **Ingresses and Load Balancing > Services**.

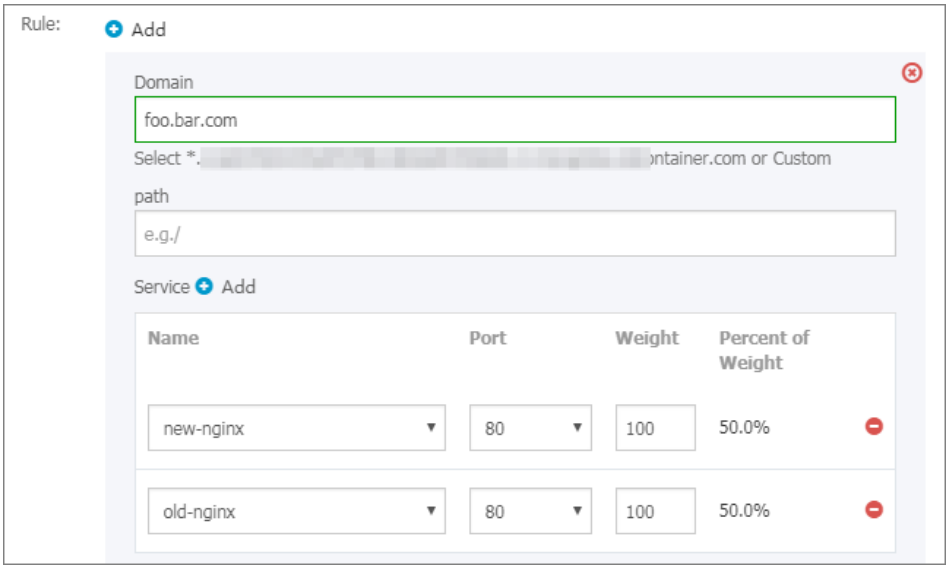
   On the Services page, find the newly created services.

Step 2: Create an Ingress

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Ingresses and Load Balancing > Ingresses**. The Ingresses page appears.
3. Select the target cluster and namespace, and click **Create** in the upper-right corner of the page.
4. In the dialog box that appears, enter the Ingress name. In this example, nginx-ingress is used.
5. Configure Ingress rules.

   Ingress rules are used to manage inbound access to the services in the cluster. The rules include HTTP or HTTPS rules. You can configure the following items in rules: domain name (virtual hostname), URL path, service name, port, and weight. For more information, see Ingress configurations.

   In this example, a rule is added to specify two services for the default domain name and virtual hostname of the cluster. Traffic routing is based on domains.



Simple fanout configurations based on domains

In this example, a virtual hostname is used as the test domain to provide services. Route weights are specified for both services and a canary release is configured for one of the services. In your production environment, you can use a domain name that has obtained an Internet Content Provider (ICP) number to provide services.

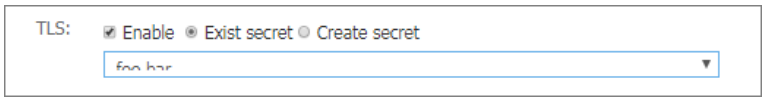
- **Domain**: Enter the test domain name. In this example, `foo.bar.com` is used.

  You must modify the hosts file to add a domain name mapping rule.

  ```
  118.178.XX.XX foo.bar.com       #The IP address of the Ingress.
  ```

- **Services**: Set the path, name, port, and weight of each service.
  - Path: Enter the URL path of each service. In this example, the default root path `/` is used.
  - Name: In this example, two service names are used: old-nginx and new-nginx.
  - Port: In this example, port 80 is opened.
  - Weight: Set the weight for each service. The weight is a percentage value. Default value: 100. In this example, the same weight 50 is set for each service.

6. Configure TLS. Select **EnableTLS** to enable TLS and configure secure routing. For more information, see Configure a secure Ingress.
   - You can select an existing secret.

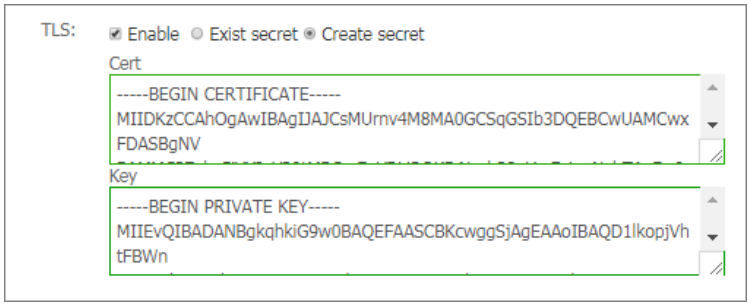
     a. Log on to a master node and create the *tls.key* and *tls.crt* files.

     ```
     openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out tls.crt -subj "/C
     N=foo.bar.com/O=foo.bar.com"
     ```

     b. Create a secret.

     ```
     kubectl create secret tls foo.bar --key tls.key --cert tls.crt
     ```

     c. On the command line, enter `kubectl get secret` and verify that the secret has been created. You can then select the *foo.bar* secret.
   - You can also use the TLS private key and certificate to create a secret.



a. Log on to a master node and create the *tls.key* and *tls.crt* files.

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out tls.crt -subj "/C
N=foo.bar.com/O=foo.bar.com"
```

b. On the command line, enter `vim tls.key` and `vim tls.crt` to obtain the private key and certificate that are generated.

c. Copy the certificate to the Cert field and the private key to the Key field.

7. Configure a canary release.

> **Note** Currently, only Ingress controllers of 0.12.0-5 and later versions support traffic splitting.

Container Service supports multiple traffic splitting methods. This allows you to select more suitable solutions for specific scenarios, such as canary releases and A/B testing, including:

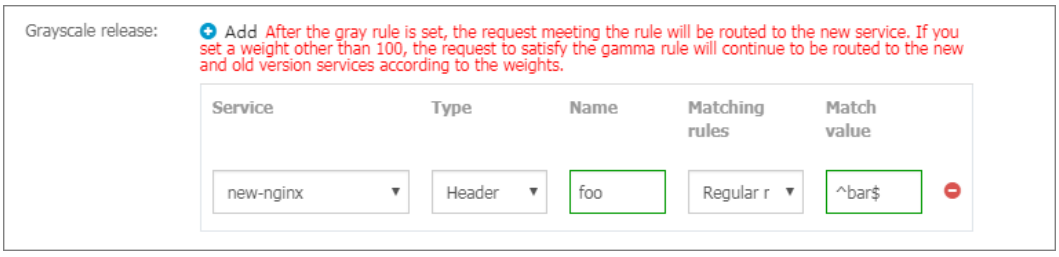
i. Traffic splitting based on request headers

ii. Traffic splitting based on cookies

iii. Traffic splitting based on query parameters

After a canary release is configured, requests that match only the specified rules are routed to the new-nginx service. If the weight of new-nginx is lower than 100%, requests that match the specified rules are routed to this service based on the weight.

In this example, the rule is added to specify a request header that matches the regular expression `foo=^bar$`. Only requests that contain this header can access new-nginx.



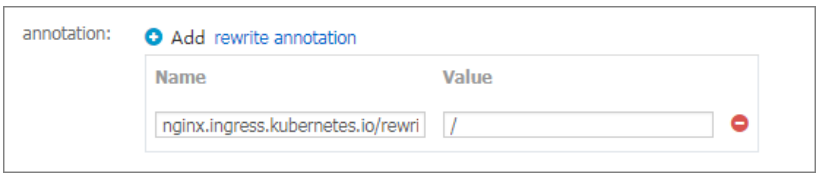
- **Services**: the service to be accessed.
- **Type**: the type of the matching rule, such as Header, Cookie, or Query.
- **Name and Match Value**: custom request fields. The name and matching value comprise a key-value pair.
- **Matching Rule**: Regular expressions and exact matches are supported.

8. Configure annotations.

Click **Rewrite Annotation** to add a rewrite annotation for the Ingress. For example, `nginx.ingress.kubernetes.io/rewrite-target: /` indicates that */path* is redirected to the root path */*. The root path can be recognized by the backend service.

> **Note** In this example, no path is configured for the service. You do not need to configure rewrite annotations. Rewrite annotations allow the Ingress to forward traffic through root paths to the backend service. This avoids the error 404 that is caused by invalid paths.

You can also click **Add** to enter annotation names and values in key-value pairs. For more information about Ingress annotations, visit https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/annotations/.

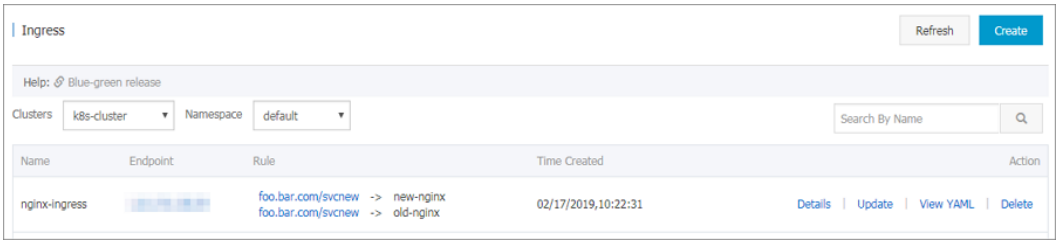


9. Add labels.

   Add labels to describe the features of the Ingress.

10. Click **Create**.

    You can find the nginx-ingress Ingress on the Ingresses page.



11. Click foo.bar.com to view the NGINX welcome page.

    When you click the domain name that points to new-nginx, the old-nginx service page appears.

> **Note** By default, when you enter the route address in the browser, requests with headers that do not contain `foo=^bar$` are directed to old-nginx.

12. Use SSH to log on to a master node. Run the following commands to simulate requests with specific headers and check the results:

```
curl -H "Host: foo.bar.com" http://47.107.XX.XX
old
 curl -H "Host: foo.bar.com" http://47.107.XX.XX
old
 curl -H "Host: foo.bar.com" http://47.107.XX.XX                    #Similar to a browser reques
t.
old
 curl -H "Host: foo.bar.com" -H "foo: bar" http://47.107.XX.XX          #Simulate a request with
a specific header. The results are returned based on the weight.
new
 curl -H "Host: foo.bar.com" -H "foo: bar" http://47.107.XX.XX
old
 curl -H "Host: foo.bar.com" -H "foo: bar" http://47.107.XX.XX
old
 curl -H "Host: foo.bar.com" -H "foo: bar" http://47.107.XX.XX
new
```

3.1.4.8.7. Update an ingress

You can update an ingress through the console.

Prerequisites

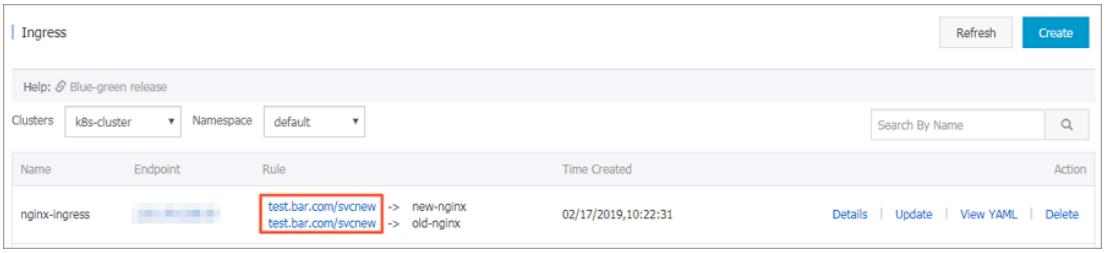
- You have created a Kubernetes cluster and an ingress controller is running normally in the cluster. For more information about creating clusters, see Create a Kubernetes cluster.
- You have created an ingress. For more information, see Create an ingress through the console.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Ingresses and Load Balancing > Ingresses**. The Ingresses page appears.
3. Select the target cluster and namespace. Find the ingress that you want to update and click **Update** in the Actions column.
4. In the dialog box that appears, modify the parameters and click **OK**. This example changes `foo.bar.com` to `test.bar.com`.

What's next

On the Ingresses page, you can find the updated ingress rule.



3.1.4.8.8. Delete an ingress

Prerequisites

- You have created a Kubernetes cluster and an ingress controller is running normally in the cluster. For more information about cluster creation, see Create a Kubernetes cluster.

- You have created an ingress. For more information, see Create an ingress through the console.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Ingresses and Load Balancing > Ingresses**.
3. Select the target cluster and namespace. Find the ingress that you want to delete and click **Delete** in the Actions column.
4. In the dialog box that appears, click **OK**.

3.1.4.9. Config maps and secrets

3.1.4.9.1. Create a ConfigMap

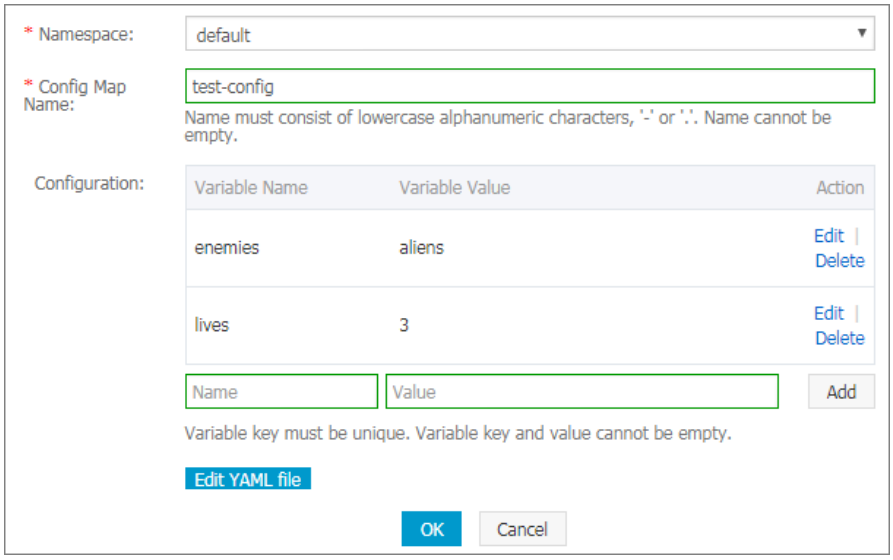
You can create a ConfigMap on the ConfigMaps page or by using a template.

Create a ConfigMap on the ConfigMaps page

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Configuration > ConfigMaps**. The ConfigMaps page appears.
3. Select the target cluster and namespace, and then click **Create**.
4. Set the parameters and click **OK**. Create a ConfigMap on the ConfigMaps page

Parameter	Description
Cluster	The ID of the selected cluster.
Namespace	The selected namespace. A ConfigMap is a kind of Kubernetes resource object and must be scoped into a namespace.
ConfigMap Name	Required. The name can contain lowercase letters, digits, hyphens (-), and periods (.). Other resource objects need to reference ConfigMap names to obtain configuration information.
ConfigMap	Enter the **Name** and **Value**, and then click **Add** to add the key-value pair. You can also click **Edit YAML file**, modify the parameters in the dialog box that appears, and then click **OK**.

In this example, two variables named enemies and lives are created. Their values are set to aliens and 3 respectively.



5. Click **OK**. You can find the newly created ConfigMap on the ConfigMaps page.

You can also click **Browse** to upload a configuration file to create a ConfigMap.

Create a ConfigMap from a template

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Applications > Deployments**. The Deployments page appears.
3. In the upper-right corner, click **Create from Template**.
4. On the page that appears, set the parameters and click **Create**. Create a ConfigMap from a template

Parameter	Description
Cluster	The cluster where the ConfigMap is created.
Namespace	The namespace where the ConfigMap belongs. A ConfigMap is a kind of Kubernetes resource object and must be scoped into a namespace.
Sample Template	Container Service provides various YAML templates for different types of resources. This helps you deploy resources quickly. You can choose *Custom* and enter your own ConfigMap based on YAML syntax, or select the *Resource-ConfigMap* template. In the sample template, the ConfigMap is named aliyun-config and contains two variable files *game.properties* and *ui.properties*. You can modify the ConfigMap based on your needs.
Template	Enter the template content based on YAML syntax. The template can contain multiple resource objects that are separated by `---`.
Add Deployment	This feature allows you to quickly define a YAML template. You can click **Use Existing Template** to import an existing template.

You can find the newly created ConfigMap *aliyun-config* on the ConfigMaps page.

3.1.4.9.2. Use a ConfigMap in a Pod

You can use a ConfigMap in a Pod in the following scenarios:

- Use a ConfigMap to define environment variables
- Use a ConfigMap to configure command line parameters
- Use a ConfigMap in volumes

For more information, see Configure a Pod to use a ConfigMap.

Limits

To use a ConfigMap in a Pod, make sure that the ConfigMap and Pod are in the same cluster and namespace.

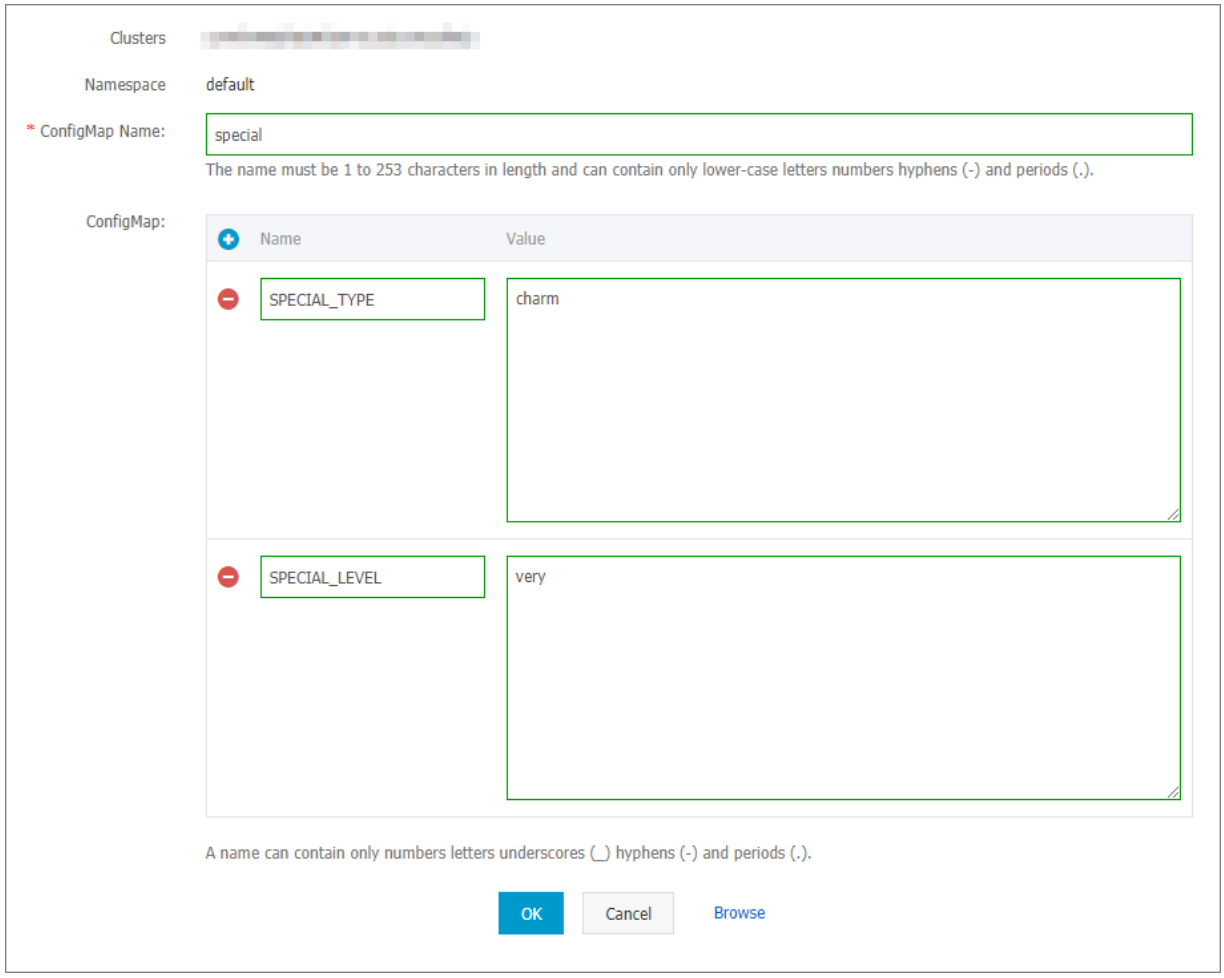
Create a ConfigMap

This example creates a ConfigMap named special_config, which consists of two key-value pairs: `SPECIAL_LEVEL: very` and `SPECIAL_TYPE: charm`.

You can use the following YAML template to create a ConfigMap.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: special-config
  namespace: default
data:
  SPECIAL_LEVEL: very
  SPECIAL_TYPE: charm
```

You can also log on to the Container Service console and choose **Configuration > ConfigMaps** in the left-side navigation pane. You can then click **Create** to create a ConfigMap.



Use ConfigMaps to define Pod environment variables

Define the value of a ConfigMap as an environment variable

You can log on to the Container Service console and choose **Applications > Deployments** in the left-side navigation pane. Click **Create from Template**, select and modify the Pod type template, and deploy the application. You can also go to the Kubernetes dashboard and choose **Upload YAML or JSON File**.

The following sample template creates a Pod and defines environment variables in the Pod. valueFrom is used to reference the value of SPECIAL_LEVEL to define an environment variable.

```
apiVersion: v1
kind: Pod
metadata:
  name: config-pod-1
spec:
  containers:
    - name: test-container
      image: busybox
      command: [ "/bin/sh", "-c", "env" ]
      env:
        - name: SPECIAL_LEVEL_KEY
          valueFrom:              ##Use valueFrom to denote that env references the value of a Config
Map.
            configMapKeyRef:
              name: special-config              ##The referenced ConfigMap name.
              key: SPECIAL_LEVEL                ##The referenced ConfigMap key.
  restartPolicy: Never
```

To define the values of multiple ConfigMaps as environment variables, you only need to add multiple env parameters in the Pod definition.

Define the key-value pairs of a ConfigMap as environment variables

To define the key-value pairs of a ConfigMap as Pod environment variables, you can use the envFrom parameter. The keys in a ConfigMap are used as the names of the environment variables.

A sample template is provided as follows:

```
apiVersion: v1
kind: Pod
metadata:
  name: config-pod-2
spec:
  containers:
    - name: test-container
      image: busybox
      command: [ "/bin/sh", "-c", "env" ]
      envFrom:                ##Reference all the key-value pairs in the special-config ConfigMap.
      - configMapRef:
        name: special-config
  restartPolicy: Never
```

Use a ConfigMap to configure command line parameters

You can use ConfigMaps to configure the commands or parameter values in a container by using the environment variable replacement syntax *$(VAR_NAME)*. A sample template is provided as follows:

```
apiVersion: v1
kind: Pod
metadata:
  name: config-pod-3
spec:
  containers:
    - name: test-container
      image: busybox
      command: [ "/bin/sh", "-c", "echo $(SPECIAL_LEVEL_KEY) $(SPECIAL_TYPE_KEY)" ]
      env:
        - name: SPECIAL_LEVEL_KEY
          valueFrom:
            configMapKeyRef:
              name: special-config
              key: SPECIAL_LEVEL
        - name: SPECIAL_TYPE_KEY
          valueFrom:
            configMapKeyRef:
              name: special-config
              key: SPECIAL_TYPE
  restartPolicy: Never
```

Run the Pod and the output is as follows:

```
very charm
```

Use a ConfigMap in volumes

You can use a ConfigMap to define volumes. The following sample template specifies a ConfigMap name under volumes. This stores the key-value pair data to the mountPath path, which is /etc/config in this example. This generates configuration files that are named after the keys of the ConfigMap. The corresponding values of the ConfigMap are stored in these files.

```
apiVersion: v1
kind: Pod
metadata:
  name: config-pod-4
spec:
  containers:
    - name: test-container
      image: busybox
      command: [ "/bin/sh", "-c", "ls /etc/config/" ]   ##List the names of files under this director
y.
      volumeMounts:
        - name: config-volume
          mountPath: /etc/config
   volumes:
     - name: config-volume
       configMap:
         name: special-config
  restartPolicy: Never
```

Run the Pod and the keys of the ConfigMap are output:

```
SPECIAL_TYPE
SPECIAL_LEVEL
```

3.1.4.9.3. Update a ConfigMap

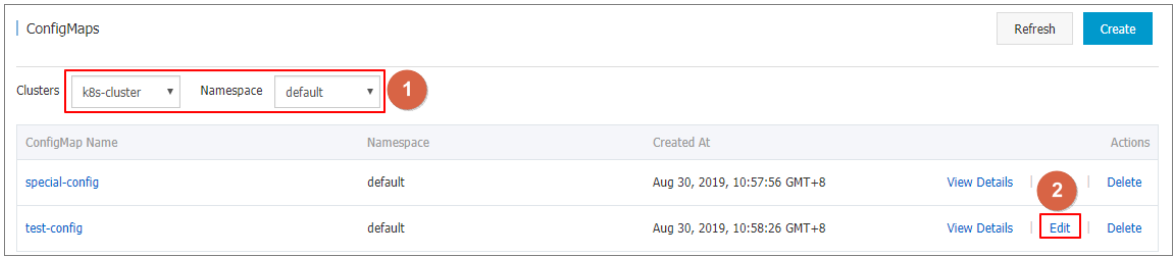
You can use multiple methods to update a ConfigMap.

Note

If you update a ConfigMap, the applications that use this ConfigMap will be affected.

Update a ConfigMap on the ConfigMaps page

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Configuration > ConfigMaps** to go to the ConfigMaps page.
3. Select the target cluster and namespace, find the ConfigMap that you want to update, and then click **Edit** in the Actions column for the ConfigMap.



4. In the dialog box that appears, modify the configurations, and click **OK**.

Update a ConfigMap by using the Kubernetes dashboard

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters** to go to the Clusters page. Find the cluster that you want to manage and click **Dashboard** in the Actions column for the cluster.
3. On the Overview page, choose **Config and Storage > ConfigMaps** in the left-side navigation pane. Select the target ConfigMap and choose  **> View/edit YAML**.
4. In the dialog box that appears, modify the configurations and click **Update**.

3.1.4.9.4. Delete a ConfigMap

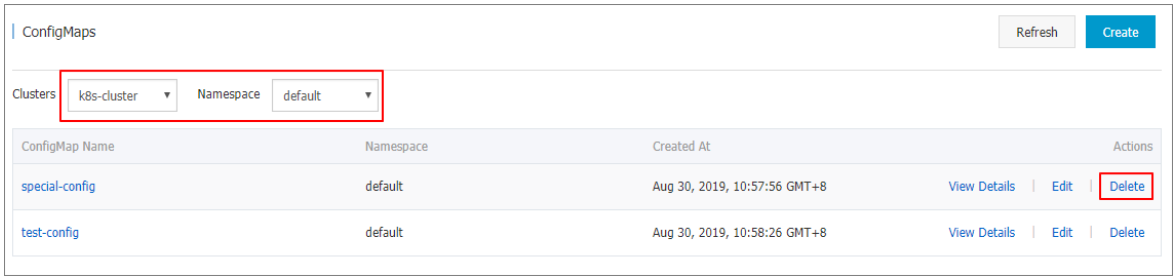
You can use multiple methods to delete a ConfigMap.

Notes

If you delete a ConfigMap, the applications that use this ConfigMap will be affected.

Delete a ConfigMap on the ConfigMaps page

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Configuration > ConfigMaps**. The ConfigMaps page appears.
3. Select the target cluster and namespace. Find the ConfigMap that you want to delete and click **Delete** in the Actions column.



Delete a ConfigMap through Kubernetes Dashboard

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters**. Find the target cluster and click **Dashboard** in the Actions column.
3. On the Kubernetes Dashboard page, choose **Config and Storage > Config Maps** in the left-side navigation pane. Find the target ConfigMap and choose **the More icon > Delete**.
4. In the dialog box that appears, click **DELETE**.

3.1.4.9.5. Create a secret

You can create secrets for applications in the Container Service console.

Prerequisites

A Kubernetes cluster is created.

Context

We recommend that you use secrets to store sensitive information in Kubernetes clusters, such as passwords and certificates.

Secrets are classified into the following types:

- Service Account: This type of secret is automatically created by Kubernetes and is automatically mounted to the pod directory */run/secrets/kubernetes.io/serviceaccount*. You can use this type of secret to access the Kubernetes API.
- Opaque: This type of secret is encoded in the Base64 format and used to store sensitive information, such as passwords and certificates.

You can use the Container Service console to create only Opaque secrets. Opaque data belongs to the map type. The value of this type must be encoded in the Base64 format. You can encode plain text in the Base64 format by using the console.

You can also manually create secrets by using the command-line interface (CLI). For more information, see Kubernetes secrets.

Procedure

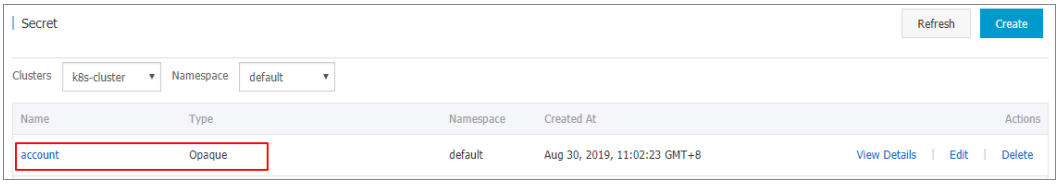
1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Configuration > Secrets**. The Secrets page appears.
3. Select the target cluster and namespace. Click **Create** in the upper-right corner of the page.
4. Configure the secret and click **OK**.

   > **Note** To enter secret data in plain text, select **Encode Data Values Using Base64**.

   Secret parameters

Parameter	Description
Name	The name of the secret that you want to create. The name must be 1 to 253 characters in length and can only contain lowercase letters, digits, hyphens (-), and periods (.).
Data	The data stored in the secret. Click **Add**, and in the dialog box that appears, enter the key and value as a key-value pair. In this example, two entries are entered: `username: admin` and `password: 1f2d1e2e67df`.

5. You can view the newly created secret on the Secrets page.



3.1.4.9.6. Edit a secret

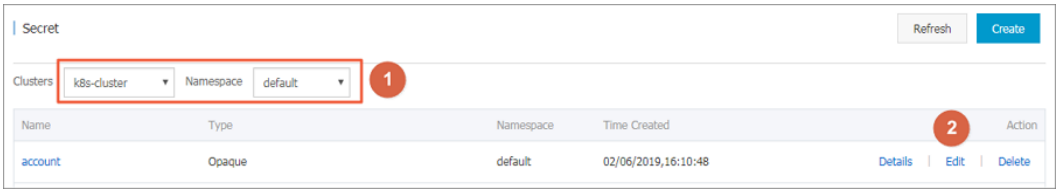
This topic describes how to edit a secret in the Container Service console.

Prerequisites

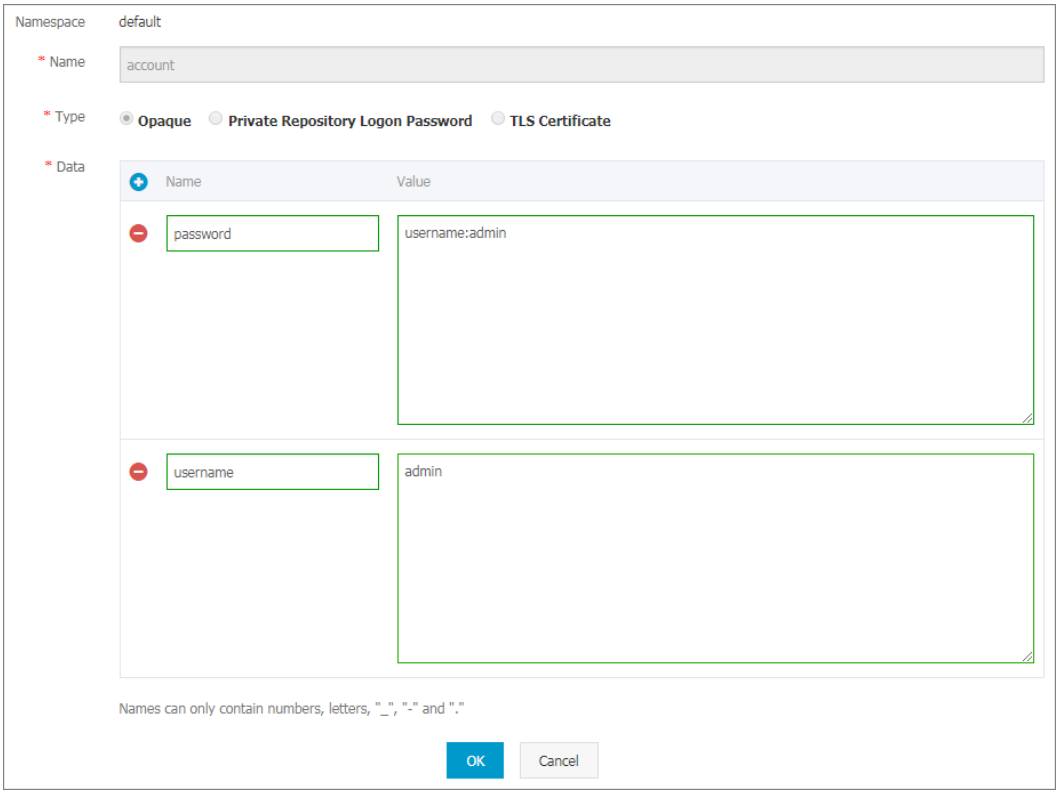
- You have created a Kubernetes cluster.
- You have created a secret. For more information, see Create a secret.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Configuration > Secrets** to go to the Secrets page.
3. Select the cluster and namespace. Choose the target secret and click **Edit** in the Actions column.



4. On the Edit Secret page, edit the secret based on your needs.



5. Click **OK** to save your edits.

3.1.4.9.7. Delete a secret

This topic describes how to delete a secret in the Container Service console.

Prerequisites

- You have created a Kubernetes cluster.
- You have created a secret. For more information, see Create a secret.

Context

> **Note** Do not delete secrets that were generated during the cluster creation process.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Configuration > Secrets** to go to the Secrets page.
3. Select the cluster and namespace. Choose the target secret and click **Delete** in the Actions column.
4. In the dialog box that appears, click **OK** to delete the secret.

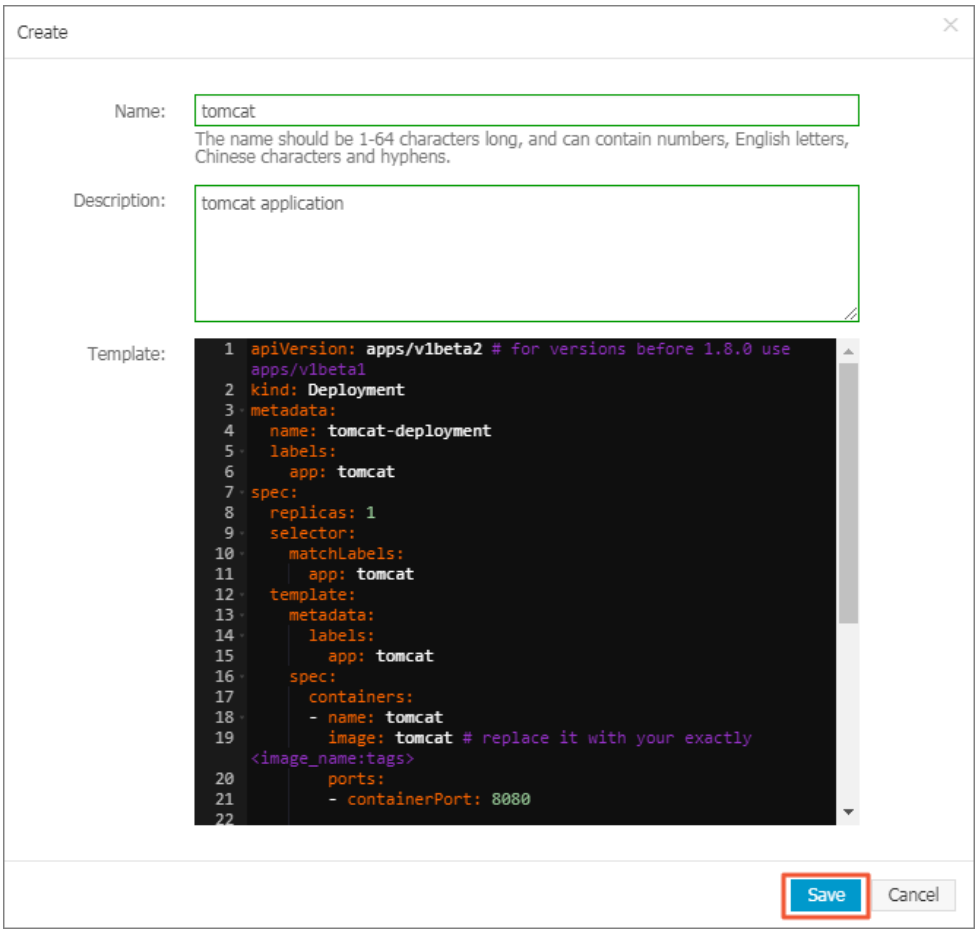
3.1.4.10. Templates

3.1.4.10.1. Create an orchestration template

This topic describes how to use multiple methods to create orchestration templates through the Container Service console.

Procedure

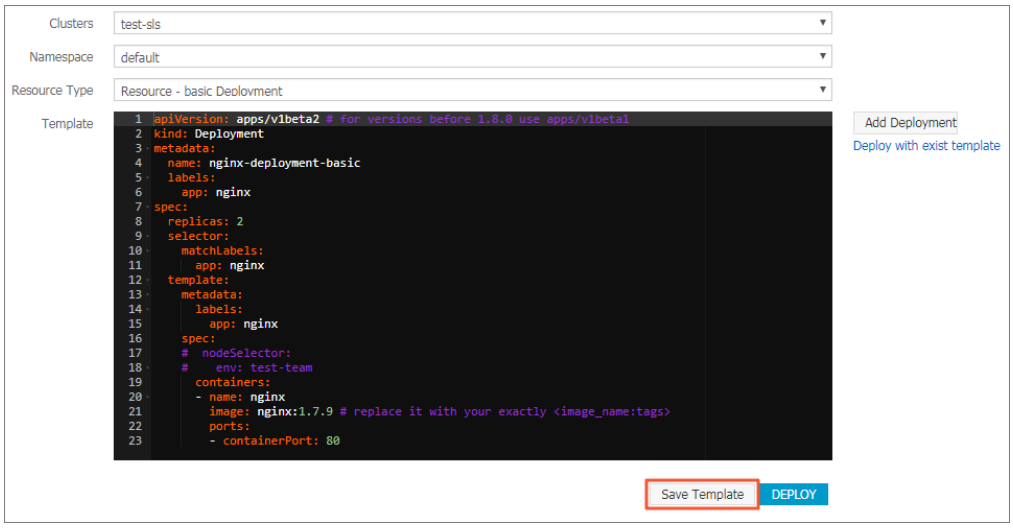
1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Marketplace > Orchestration Templates** and click **Create** in the upper-right corner.
3. In the dialog box that appears, configure the template, and then click **Save**. This example demonstrates how to create a Tomcat application template that contains a deployment and a service.
   - **Name**: The name of the template.
   - **Description**: Optional. The description of the template.
   - **Template**: Enter the template content based on YAML syntax. The template can contain multiple resource objects that are separated by `---`.



4. After the template is created, you are redirected to the **Templates** page by default. You can find the template on the **My Templates** tab.

5. (Optional)You can also choose **Applications > Deployments** in the left-side navigation pane, and click **Create from Template** to go to the **Create from Template** page. You can modify a built-in template provided by Container Service and save it as a custom template.

i. Select a built-in template and click **Save Template**.



ii. In the dialog box that appears, specify the name, description, and content. Click **Save** to save the template.

> **Note** You can modify the built-in template based on your needs.

iii. In the left-side navigation pane, choose **Marketplace > Orchestration Templates**. You can find the newly created template on the **My Templates** tab.

What's next

You can use the orchestration templates on the **My Templates** tab to quickly create applications.

3.1.4.10.2. Update an orchestration template

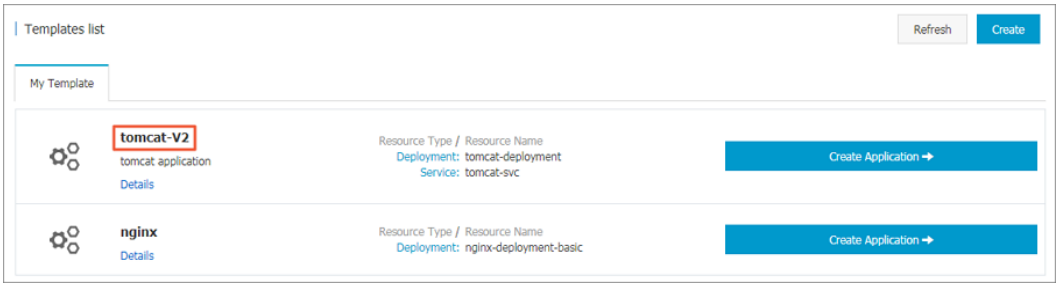
This topic describes how to edit and update an orchestration template.

Prerequisites

You have created an orchestration template. For more information, see Create orchestration templates.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Marketplace > Orchestration Templates**. The **Templates** page appears. You can view existing templates on the **My Templates** tab.
3. Select the target template and click **Details**.
4. On the template details page, click **Edit** in the upper-right corner.
5. In the dialog box that appears, edit the name, description, and template content, and click **Save**.
6. Go to the **Templates** page. You can view the template that you have updated on the **My Templates** tab.



3.1.4.10.3. Save an orchestration template as a new one

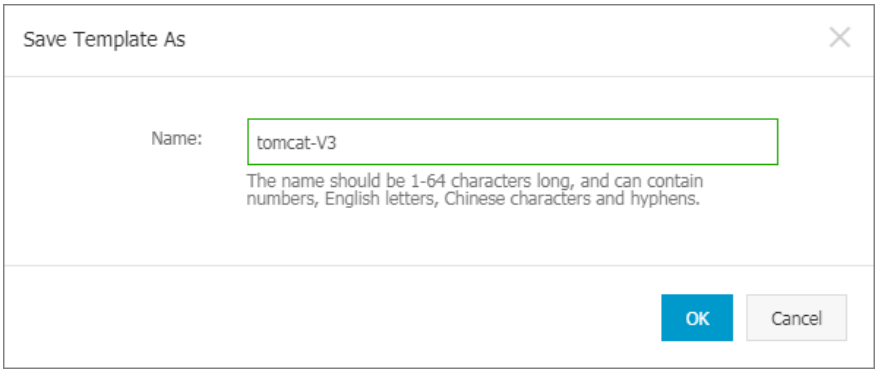
This topic describes how to save an orchestration template as a new one.

Prerequisites

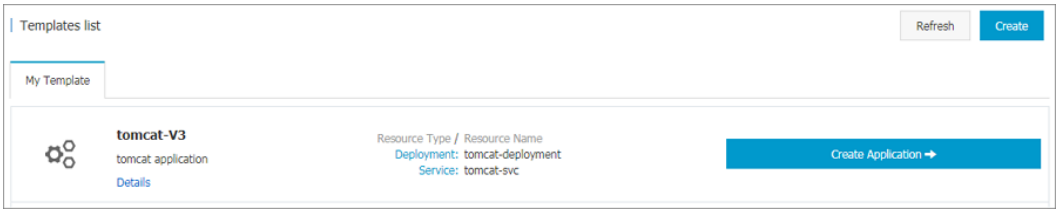
You have created an orchestration template. For more information, see Create orchestration templates.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Marketplace > Orchestration Templates**. The **Templates** page appears. You can view existing templates on the **My Templates** tab.
3. Select the target template and click **Details**.
4. On the template details page, modify the template and click **Save As** in the upper-right corner.
5. In the dialog box that appears, enter the template name and click **OK**.



6. Go to the **Templates** page. The newly saved template is displayed on the **My Templates** tab.



3.1.4.10.4. Download an orchestration template

This topic describes how to download an orchestration template.

Prerequisites

You have created an orchestration template. For more information, see Create orchestration templates.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Marketplace > Orchestration Templates**. The **Templates** page appears. You can view existing templates on the **My Templates** tab.
3. Select the target template and click **Details**.
4. On the template details page, click **Download** in the upper-right corner to download the template as a YAML file.

3.1.4.10.5. Delete an orchestration template

Prerequisites

You have created an orchestration template. For more information, see Create orchestration templates.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Marketplace > Orchestration Templates**. The **Templates** page appears. You can view existing templates on the **My Templates** tab.
3. Select the target template and click **Details**.
4. On the template details page, click **Delete** in the upper-right corner.
5. In the dialog box that appears, click **OK**.

3.1.4.11. Images

3.1.4.11.1. Create an image repository

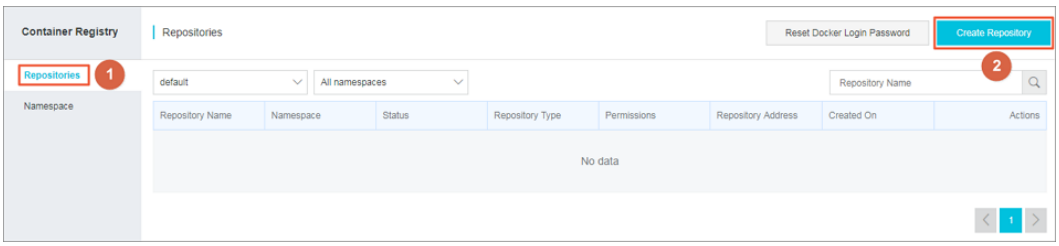
This topic describes how to create an image repository in the Container Registry console.

Prerequisites

The Container Registry service is activated and a namespace is created in the Container Registry service.

Procedure

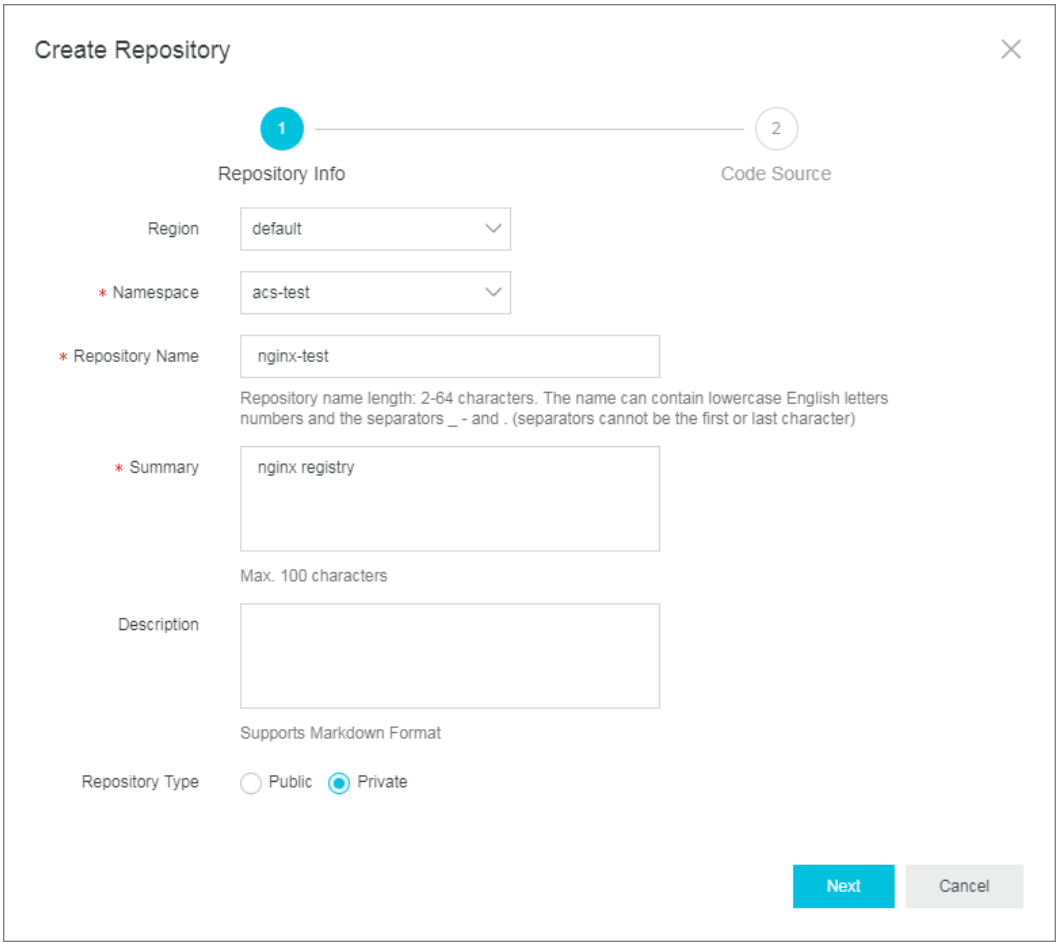
1. Log on to the Container Registry console.
2. In the left-side navigation pane, click **Repositories** to go to the Repositories page. Click **Create Repository** in the upper-right corner of the page.



3. In the dialog box that appears, set the following parameters and click **Next**.
   - **Region**: Use the default value. The region of the repository must be the same as that of the cluster.
   - **Namespace**: Use the default value. The namespace must be the same as that of the organization that you select when you log on to the console.
   - **Repository Name**: The repository name must be 2 to 64 characters in length and can contain lowercase letters, digits, and special characters, including underscores (_), hyphens (-), and periods (.). It cannot start

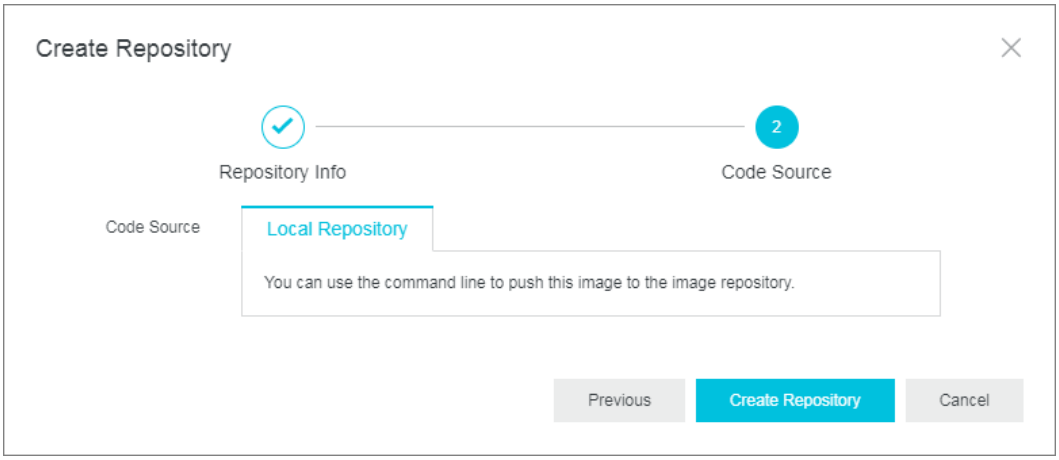
or end with a special character.

- **Summary**: Enter the repository summary.
- **Description**: Enter description information. The description must be 0 to 100 characters in length.
- **Repository Type**: The repository type can be public or private.

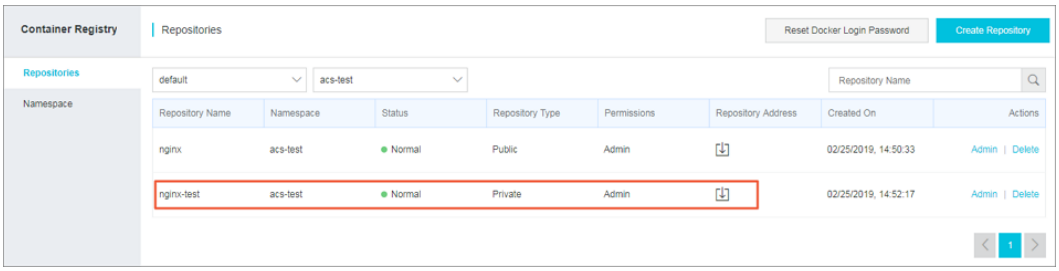


4. Set the code source and click **Create Repository**.

> **Note** Currently, only local repositories are supported. You can push images to an image repository by using the command-line interface (CLI).

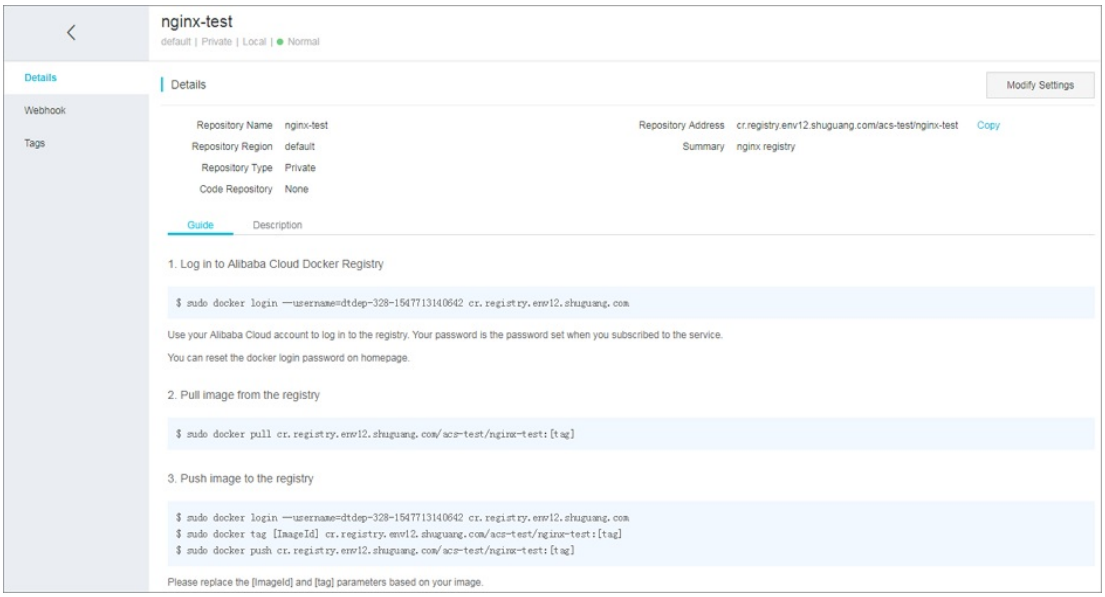


5. Go to the Repositories page to view the newly created repository.



What's next

You can click **Admin** in the **Actions** column for the repository to go to the **Details** page and learn how to manage the repository.



3.1.4.11.2. Create a namespace

This topic describes how to create a namespace in the Container Registry console.

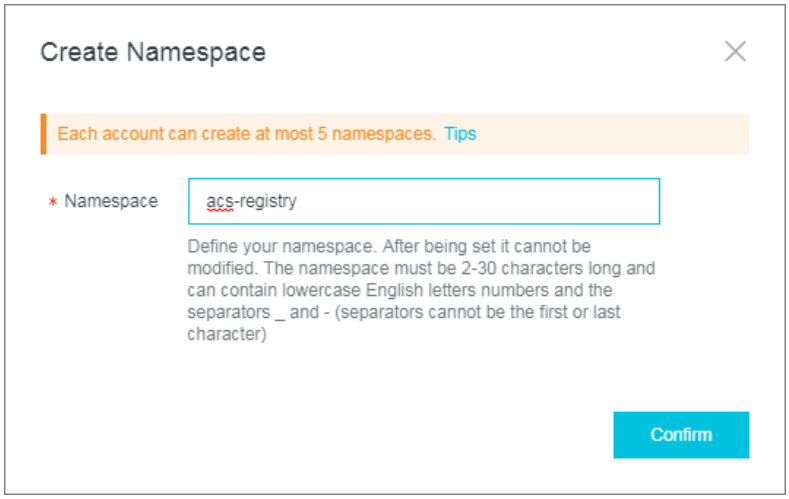
Context

A namespace is a collection of repositories. We recommend that you place the repositories of a company or an organization in the same namespace.

- You can enter a name that corresponds to a company name, such as aliyun or alibaba.
- You can enter a name that corresponds to a team or organization name for the namespace, such as misaka-team.

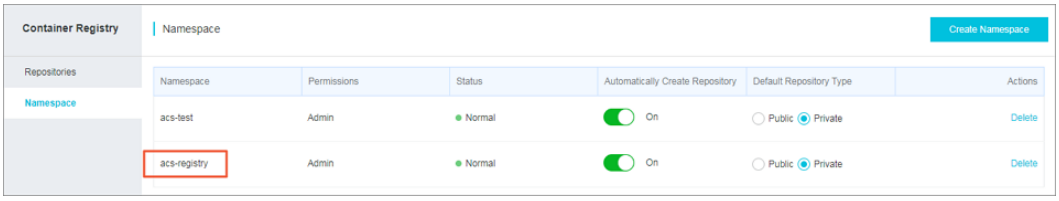
Procedure

1. Log on to the Container Registry console
2. In the left-side navigation pane, click **Namespace**. On the page that appears, click **Create Namespace** to create a namespace.



You can find the newly created namespace on the Namespace page.

> **Note**
> - You can turn on or turn off **Automatically Create Repository** to specify whether to automatically create image repositories for the namespace.
> - You can set the repository type to **Public** or **Private** for the namespace.



What's next

You can create more image repositories for the namespace.

3.1.4.11.3. Synchronize an image

Container Registry can be used to synchronize images across data centers. This allows you to retrieve images from the data center that is nearest to where your workloads are deployed.

Prerequisites

- A namespace is created. For more information, see Create a namespace.
- An image repository for the namespace is created. For more information, see Create a repository.

Procedure

1. Log on to the Container Registry console.
2. In the left-side navigation pane, choose **Repositories** to go to the **Repositories** page.
3. Select the target repository and click **Admin** in the **Actions** column for the repository to go to the Details page.
4. In the left-side navigation pane, choose **Tags**. Find the target tag and click **Sync** in the **Actions** column for the tag.
5. On the **Sync** page, specify the tag and target repository, and click **OK**.

   A message appears to indicate that the synchronization request is submitted.

Result

In the left-side navigation pane of the target repository, click **Sync** to check the status of the synchronization task.

You can click **Details** in the Actions column for the task to check the task status for the specified image.

3.1.4.11.4. Sign and verify an image

When you manage container images, you can use content trust to verify both the integrity and the publisher of images. Image publishers can encrypt images by using digital signatures that are stored in Container Registry. We recommend that you verify the signatures to ensure that only images signed by trusted authorities are deployed. This way, you can minimize or stop attacks that may occur when you run containers.

Install and configure signature tools

1. Install the aliyun client tool. For more information, see Install Alibaba Cloud CLI.
2. On the command line, run the following command to configure aliyun client:

```
./aliyun configure set \
  --profile akProfile \
  --mode AK \
  --region cn-qingdao-env17-d01 \
  --access-key-id yourAK \
  --access-key-secret yourAK
```

3. After you install and configure the GPG tool, run the following command to export the public key:

```
#Query the pubKeyId.
gpg --fingerprint
#Export the public key.
gpg --armor --output public-key.txt --export yourUser
```

> **Note** For Apsara Stack instances, instanceId is set to default.

Sign container images

1. On the command line, run the following command to determine the image tag URL:

```
echo image://region/instanceId/namespace/repo@digest | tee imageURL.txt
#e.g.
#echo image://cn-qingdao-env17-d01/default/kritis-test/busybox2@sha256:2f122941b5850006dbb7adda78
d2ea5b382841ca6569fd174bd24c14bfff3dca > imageURL.txt
```

2. On the command line, run the following command to use GPG to sign the unique URL of the image. By default, imageURL.txt.asc is generated.

```
gpg --armor --sign imageURL.txt
```

The imageURL.txt.asc file includes the following content:

```
-----BEGIN PGP MESSAGE-----
owGbwMvMwMEo63vGqaX74wXGNXVJ3CmpaYmlOSV6JRUlcZy7vmfmJqanWunr****
.......
=O2TP
-----END PGP MESSAGE-----
```

3. On the command line, run the following command to create a metadata namespace:

```
./aliyun cr CreateMetadataNamespace --force --version 2018-12-01 --endpoint cr.inter.env17e.shuguang.com --NamespaceName kritis-test-2 --Description "for test"
```

4. On the command line, run the following command to create a signature note:

```
cat <<EOF > note.json| jq
{
  "name": "/namespaces/kritis-test-2/notes/image-sign",
  "LongDescription": "long",
  "ShortDescription": "short",
  "ExpirationTime": "2021-01-01T00:00:00Z",
  "Kind": "ATTESTATION",
  "Attestation": {
    "Hint": {
      "HumanReadableName": "ACR"
    }
  }
}
EOF
```

5. On the command line, run the following command to create an occurrence for the image signature and save the occurrence to the metadata service:

```
cat <<EOF > occurrence.json | jq
{
  "Name": "/namespaces/kritis-test-2/occurrences/randomId1",
  "NoteName": "/namespaces/kritis-test-2/notes/image-sign",
  "ResourceUri": "image://cn-qingdao-env17-d01/default/kritis-test/busybox2@sha256:2f122941b5850006dbb7adda78d2ea5b382841ca6569fd174bd24c14bff****",
  "Kind": "ATTESTATION",
  "Attestation": {
    "Signatures": [
      {
        "Signature": $(cat imageURL.txt.asc|jq -R --slurp),
        "PublicKeyId": "E5B5FF2AFC3A1D70FE3CE57C1D4DCC42848****"
      }
    ]
  }
}
EOF
```

```
./aliyun cr CreateMetadataOccurrence --force --version 2018-12-01 --endpoint cr.inter.env17e.shuguang.com --NamespaceName kritis-test-2 --OccurrenceName randomId1 --Occurrence "$(cat occurrence.json)"
```

6. On the command line, run the following command to query the image signature:

```
#Retrieve a list of occurrences for a specified note. In this example, one occurrence is created.
./aliyun cr ListMetadataOccurrences --force --version 2018-12-01 --endpoint cr.inter.env17e.shuguang.com --NamespaceName kritis-test-2 --NoteName image-sign --PageNo 1 --PageSize 5
#Retrieve a list of occurrences for a specified note that occurs on a resource. In this example, one occurrence is created.
./aliyun cr ListMetadataOccurrences --force --version 2018-12-01 --endpoint cr.inter.env17e.shuguang.com --NamespaceName kritis-test-2 --NoteName image-sign --PageNo 1 --PageSize 5 --ResourceURIs '["image://cn-qingdao-env17-d01/default/kritis-test/busybox2@sha256:2f122941b5850006dbb7adda78d2ea5b382841ca6569fd174bd24c14bfff****"]'
```

Verify an image signature

The following steps describe how to install an image signature verification component and enable the verification feature on Apsara Stack.

1. Install the image signature verification component.
   i. Log on to the Container Service console.
   ii. Go to the Clusters page, find the cluster in which you want to install the image signature verification component, and in the Actions column for the cluster, choose **More** > Install Kritis.
2. Configure the signature verification policy.
   i. Set the following signature parameters to configure the signature verification policy:

```
$ export namespace=Actual value of namespace
$ export noteName=Actual value of noteName
$ export publicKeyData=Actual value of publicKeyData
```

   - namespace: the ACR namespace setting used for signing an image.
   - noteName: the noteName setting used for signing an image.
   - publicKeyData: the Base64-encoded GPG public key.

   ii. On the command line, run the following command to set AttestationAuthority:

> **Note** The default namespace is used in the following example.

```
$ cat <<EOF > aa.yaml
apiVersion: kritis.grafeas.io/v1beta1
kind: AttestationAuthority
metadata:
  name: ${noteName}
spec:
  noteReference: namespaces/${namespace}
  publicKeyData: ${publicKeyData}
EOF
$ kubectl -n default apply -f aa.yaml
```

   iii. On the command line, run the following command to set GenericAttestationPolicy:

> **Note** The default namespace is used in the following example.

```
$ cat <<EOF > gap.yaml
apiVersion: kritis.grafeas.io/v1beta1
kind: GenericAttestationPolicy
metadata:
  name: my-gap
spec:
  attestationAuthorityNames:
  - ${noteName}
EOF
$ kubectl -n default apply -f gap.yaml
```

3. Test the signature verification feature.

   In the test, a signed image is used: registry.acs.example.com/kritis-test/signed@sha256:2f122941b5850006dbb7adda78d2ea5b382841ca6569fd174bd24c14bfff****. An unsigned image is also used: registry.acs.example/kritis-test/not-sign@sha256:efc961b2b3499c25753d3c9f29977f494f49125cf1191071057aa68bffa7****. If the feature functions as expected, the signature verification policy enables the signed image and disables the unsigned image for the default namespace. The following example shows how to test the feature:

```
#A deployment is created based on the signed image.
$ kubectl -n default run test-signed --image=registry.acs.example.com/kritis-test/signed@sha256:2
f122941b5850006dbb7adda78d2ea5b382841ca6569fd174bd24c14bff****
deployment.apps/test-signed created
#The unsigned image fails a deployment.
$ kubectl -n default run test-not-signed --image=registry.acs.example/kritis-test/not-sign@sha256
:efc961b2b3499c25753d3c9f29977f494f49125cf1191071057aa68bffa7****
Error from server: admission webhook "kritis-validation-hook-deployments.grafeas.io" denied the r
equest: image registry.acs.example/kritis-test/not-sign@sha256:efc961b2b3499c25753d3c9f29977f494f
49125cf1191071057aa68bffa7**** is not attested
```

Note After you enable image signature verification, when you create resources, the image ID must be a digest value in the format such as @sha256:<hash>.

Appendix 1: Use GPG commands to generate publicKeyData

1. On the command line, run the following command to find the local GPG key user that you want to use:

   Note The user in the following example is `abcdef@example.com`.

   ```
   $ gpg --list-keys
   pub   rsa2048 2020-01-08 [SC] [Expired on: 2022-01-07]
         7726310BC6E11E9B57B9CC08E2932E4363F3***
   uid [Absolute] abcdef <abcdef@example.com>
   sub   rsa2048 2020-01-08 [E] [Expired on: 2022-01-07]
   ```

2. Export the public key of this user and encode the content of the public key in the Base64 format to generate publicKeyData.

   ```
   $ gpg --armor --export  <user> |base64 | tr -d '\n'
   #In the example, the user is abcdef@example.com.
   #Run this command to generate publicKeyData: gpg --armor --export  abcdef@example.com |base64 | t
   r -d '\n'
   # export publicKeyData=$(gpg --armor --export  abcdef@example.com |base64 | tr -d '\n')
   ```

Appendix 2: Retrieve an image digest value

You can use the following method to find an image digest value. In this example, the image URL is registry.acs.example.com/kritis-test/alpine:3.11.

```
$ docker pull registry.acs.example.com/kritis-test/alpine:3.11
$ docker images --digests | grep registry.acs.example.com/kritis-test/alpine
registry.acs.example.com/kritis-test/alpine 3.11 sha256:ddba4d27a7ffc3f86dd6c2f92041af252a1f23a8e742c
90e6e1297bfa1bc0c45 e7d92cdc71fe 2 months ago  5.59MB
```

The output shows that the image digest value is sha256:ddba4d27a7ffc3f86dd6c2f92041af252a1f23a8e742c90e6e1297bfa1bc0c45. When you create resources, the following image ID must be used: registry.acs.example.com/kritis-test/alpine:sha256:ddba4d27a7ffc3f86dd6c2f92041af252a1f23a8e742c90e6e1297bfa1bc0c45.

3.1.4.11.5. Synchronize images between instances

You can configure synchronization rules to automatically synchronize images from a source instance to a destination instance. By default, images are automatically synchronized for instances that are deployed in different regions.

Procedure

1. Log on to the Container Registry console.
2. Go to the **Repositories** page, find the repository that you want to manage, and then click **Admin** in the **Actions** column for the repository.
3. The Details page is displayed. In the left-side navigation pane, click **Sync**.
4. In the upper-right corner of the Sync page, click **Create**. In the Sync dialog box, set the **Tag** and **Target** parameters, specify the namespace, repository name, and tag of the target repository, and then click **OK**. The synchronization rule is created.

Result

After the synchronization rule is created, a synchronization task is automatically triggered when a new image is uploaded to the repository whose name matches the specified synchronization rule.

3.1.4.12. Auto scaling

3.1.4.12.1. Auto scaling of nodes

Container Service provides the auto scaling component for nodes to automatically scale in and out. General purpose instances and GPU-accelerated instances can be automatically added to or removed from a Container Service cluster based on your requirements. This feature supports different scaling modes and is applicable to instances that are deployed across multiple zones and diverse instance types.

How it works

The auto scaling model of Container Service works in a pattern different from the traditional auto scaling model that is based on a resource usage threshold. Developers must understand these differences when they migrate workloads from data centers or other orchestration systems such as Swarm to Kubernetes.

The traditional auto scaling model works based on a resource usage threshold. For example, if a cluster has three nodes, and the CPU and memory usage of the nodes in the cluster exceeds a specific threshold, new nodes are added to the cluster. In this case, the following issues may occur:

- How is a resource usage threshold specified and applied?

  In a cluster, hot nodes may cause high resource usage and other nodes may cause low resource usage. If an average resource usage is specified as the threshold, auto scaling may be delayed. If the lowest resource usage is specified as the threshold, resources may be wasted during the scaling.

- How is load balancing applied after instances are added?

  In Container Service, a pod is used as the smallest unit that runs an application on each node of a Kubernetes cluster. If the resource usage of a pod is high, auto scaling is triggered on the node or in the cluster to which the pod belongs. If the number of pods that run the application and the limits on the pods are unchanged, the workloads cannot be distributed to the added nodes.

- How is a scale-in event triggered and implemented?

  If a scale-in event is triggered based on a resource usage threshold, the pods that request a large number of resources from nodes but have low resource usage may be evicted from the nodes. If the number of this type of pod is high in the cluster, scheduling resources may be exhausted and specific pods cannot be scheduled.

To fix these issues, Container Service provides a two-layer auto scaling model in which scheduling is decoupled from resource usage.

In this model, application replicas are modified and scheduled based on the resource usage. If pods become pending due to insufficient scheduling resources in the Kubernetes cluster, new nodes are added to the Kubernetes cluster. The pending pods are scheduled to the new nodes. This optimizes load balancing for the application that is deployed in the pods. The auto scaling model of Container Service supports the following features:

- Triggers a scale-out event

  cluster-autoscaler is used to trigger auto scaling by detecting pending pods. When pods become pending due to insufficient scheduling resources, cluster-autoscaler simulates scheduling. Then, the simulated scheduler computes which scaling group can provide new nodes to accept the pending pods. If a scaling group meets the requirements, new nodes are added in the scaling group. During the simulated scheduling, a scaling group is treated as an abstract node, and the instance types in the scaling group indicate the CPU, memory, and GPU capacities of the node. Then, labels and taints are added to the scaling group in the same way labels and taints are added to the node. The simulated scheduler references the abstract node during the simulated scheduling. If pending pods can be scheduled to the abstract node, the number of required nodes is calculated to drive the scaling group to create the required nodes.

- Triggers a scale-in event

  Only scaled-out nodes can be scaled in. Static nodes cannot be managed by cluster-autoscaler. A scale-in event is triggered on an individual node. When the scheduling resource usage of a node is lower than the specified scheduling threshold, a scale-in event is triggered on the node. In this case, cluster-autoscaler simulates pod eviction on the node and checks whether all pods can be evicted from the node. cluster-autoscaler does not drain the nodes that contain certain pods. For example, if the non-DaemonSet pods that belong to the kube-system namespace or the pods that are controlled by a pod disruption budget (PDB) run on a node, cluster-autoscaler skips this node and then choose among other candidate nodes. After all pods are evicted from a node to other nodes, the node is drained and does not serve your workloads.

- Determines a scaling group for auto scaling

  Each scaling group is regarded as an abstract node. cluster-autoscaler selects a scaling group for auto scaling based on a policy that is similar to the scheduling policy. Nodes that meet the scheduling policy are detected. Then, the nodes that meet the policies such as affinity settings are selected. If the nodes do not meet all the policies, cluster-autoscaler selects a scaling group based on the least-waste policy. The least-waste policy ensures the fewest remaining idle resources after simulation. If a GPU scaling group and a CPU scaling group can be used for auto scaling at the same time, a scale-out event is triggered in the CPU scaling group in priority.

- Improves the success rate of auto scaling based on the following factors:
  - Whether the scheduling policy is met

    After a scaling group is configured, developers must determine the scope of the scheduling policies for the pods that the scaling group supports. You can set the nodeSelector field to filter the required label of the scaling group and simulate a scale-out event.

  - Whether scaling resources are sufficient

    After the simulated scheduling is passed, a scaling group is selected to add nodes. However, whether the specified types of Elastic Compute Service (ECS) instances in the scaling group are available determines whether the instances can be deployed. Therefore, you can select different types of instances in multiple zones to improve the success rate of auto scaling.

- Increases the auto scaling speed
  - Method 1: Enable the swift mode to accelerate auto scaling. After a scaling group experiences a scale-out event and a scale-in event, the swift mode can be enabled for this scaling group.
  - Method 2: Use a custom image as a base image based on Alibaba Cloud Linux 2 (formerly known as Aliyun Linux 2). This allows you to accelerate resource delivery at the Infrastructure-as-a-Service (IaaS) layer by 50%.

Usage notes

- For each Alibaba Cloud account, the default CPU quota of pay-as-you-go instances in each region is 50 vCPUs. You can create a maximum of 48 custom route entries in each route table in a virtual private cloud (VPC). To

request a quota increase, submit a ticket.

- If you select ECS instances of the same type for auto scaling, the availability of ECS instances fluctuates. We recommend that you configure multiple instance types of the same specification in a scaling group. This improves the success rate of auto scaling.
- In Swift mode, when a node is shut down and reclaimed, the node enters the *NotReady* state. When a scale-out event is triggered, the node enters the *Ready* state.
- When a node is shut down and reclaimed in Swift mode, you are charged for only the storage costs of disks. Exceptions are nodes that use local disks, such as the instance type of ecs.d1ne.2xlarge, where the computing costs are also charged. However, if the stock of the node resources is sufficient, nodes can be launched at the earliest opportunity.
- If elastic IP addresses (EIPs) are associated with the pods, we recommend that you do not delete the auto scaling group or the ECS instances that are scaled out by the auto scaling group in the ECS console. Otherwise, the EIPs cannot be automatically released.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters**.
3. On the Clusters page, find the cluster that you want to manage and choose **More > Auto Scaling** in the **Actions** column.
4. On the **Configure Auto Scaling** page, set the following parameters and click **Submit**.

Parameter	Description
Cluster	The name of the cluster for which you want to configure auto scaling.
Scale-in Threshold	For a scaling group that is managed by cluster-autoscaler, set the value to a ratio of the requested resources on a node to the total resources on the node. If the actual value is lower than the threshold, the node is removed from the cluster. **Note** In auto scaling, a scale-out event is automatically triggered based on the node scheduling of the Kubernetes cluster. Therefore, you are required to set only scale-in parameters.
GPU Scale-in Threshold	The scale-in threshold for GPU-accelerated instances. If the actual value is lower than the threshold, the node is removed from the Kubernetes cluster.
Defer Scale-in For	The time to wait after the scale-in threshold is met and before the scale-in event starts. Unit: minutes. Default value: 10.
Cooldown	The cooldown period after a node is removed. No scale-in events are triggered again during this period. Unit: minutes. Default value: 10.

5. Click **Create Scaling Group** and specify the type of resource for auto scaling based on your business requirements. General purpose instances and GPU-accelerated instances are available.
6. In the **Configure Scaling Group** dialog box, set the following parameters.

Parameter	Description
Region	The region where the scaling group is deployed. The scaling group and the Kubernetes cluster must be deployed in the same region. You cannot change the region after the scaling group is created.
VPC	The scaling group and the Kubernetes cluster must be deployed in the same VPC.
VSwitch	The zone of the scaling group and the pod CIDR blocks that are available to the scaling group.

7. Configure worker nodes.

Parameter	Description
Instance Type	The instance types in the scaling group.
Selected Instance Specifications	The instance types that you have selected. You can select a maximum of 10 instance types.
System Disk	The system disk of the scaling group.
Attach Data Disk	Specifies whether to mount data disks to the scaling group. By default, no data disk is mounted.
Instances	The number of instances in the scaling group. **Note** ◦ Existing instances in the Kubernetes cluster are excluded. ◦ By default, the minimum number of instances is zero. When the number of instances exceeds zero, the system adds instances to the scaling group. When a scale-out event is triggered, instances in the scaling group are added to the Kubernetes cluster where the scaling group resides.
Password	Set the password. ◦ Password: Enter the password that is used to log on to the nodes. ◦ Confirm Password: Enter the password again.
Scaling Mode	You can select **Standard** or **Swift**.
RDS Whitelist	The ApsaraDB RDS instances that can be accessed by the nodes in the scaling group after a scale-out event is triggered.
Label	Node labels are automatically added to the nodes that are added to the cluster after a scale-out event is triggered.
ECS Label	You can add labels to the selected ECS instances.
Taints	After you add taints to a node, Container Service does not schedule pods to the node.

Parameter	Description
CPU Policy	Set the CPU policy. ○ none: the default CPU policy. This policy enables the default CPU affinity scheme. ○ static: This policy allows pods with specific resource characteristics on the node to be granted with enhanced CPU affinity and exclusivity.

8. Set advanced options.

Parameter	Description
Security Group	Set the security group for the Kubernetes cluster.
Custom Image	You can select a custom image. Then, all nodes in the Kubernetes cluster are deployed based on the image.
User Data	Customize the startup behaviors of ECS instances and import data to the ECS instances. The user data can be used to: ○ Run user data scripts during instance startup. ○ Pass user data as common data into an ECS instance for future reference.

9. Click **OK**.

Verify the results

In the left-side navigation pane, choose **Applications > Deployments**, select the scaled cluster and the kube-system namespace. You can find the cluster-autoscaler component. This indicates that the scaling group is created.

FAQ

- Why does the auto scaling component fail to add nodes after a scale-out event is triggered?

  Check whether the following scenarios exist:

  - The instance types in the scaling group cannot meet the requested resources of pods. By default, system components are installed for each node. Therefore, the requested pod resources must be less than the resource capacity of the instance type.
  - You do not authorize the Resource Access Management (RAM) roles to manage the Kubernetes cluster. Authorization must be performed for each Kubernetes cluster that is involved in the scale-out event.
  - The Kubernetes cluster cannot connect to the Internet. The auto scaling component must call Alibaba Cloud API operations. Therefore, the nodes must have access to the Internet.

- Why does the auto scaling component fail to remove nodes after a scale-in event is triggered?

  Check whether the following scenarios exist:

  - The requested resource threshold of each pod is higher than the configured scale-in threshold.
  - Pods in the *kube-system* namespace are running on the node.
  - A pod on the node contains a forced scheduling policy. As a result, other nodes cannot run the pod.
  - The pods on the node have PodDisruptionBudget and reach the minimum value of PodDisruptionBudget.

  For more information about FAQ, visit the open source community.

- How does the system select from multiple scaling groups for a scale-out event?

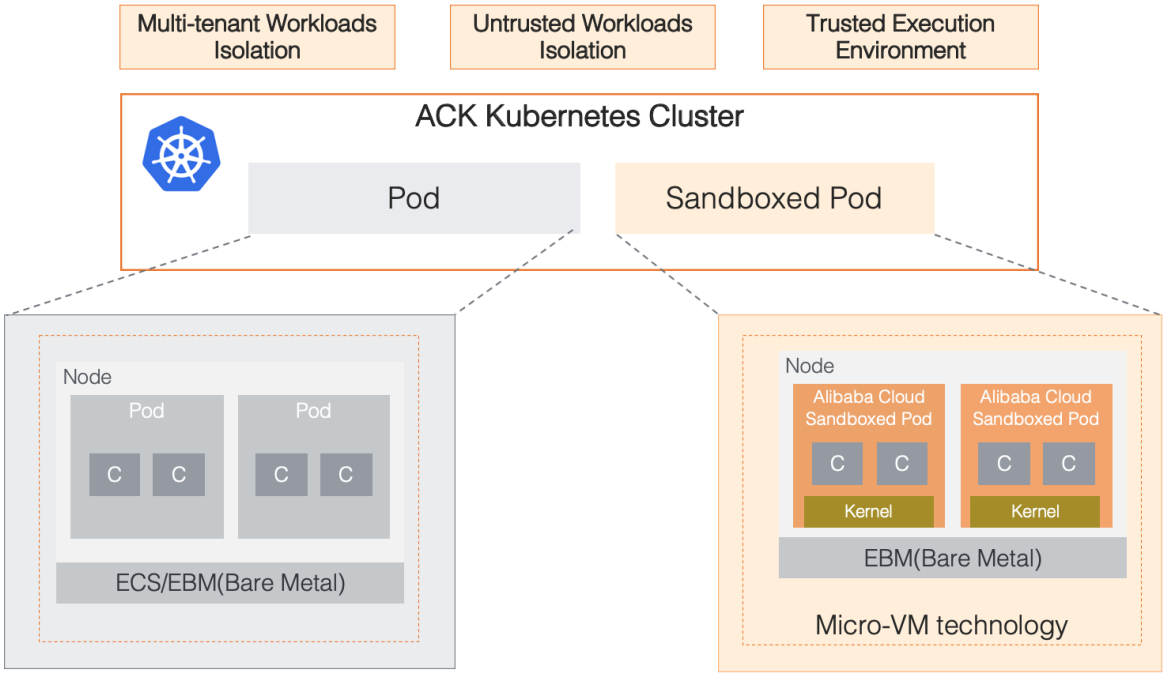
If pods cannot be scheduled to nodes, the auto scaling component simulates the scheduling of pods based on the configurations of the scaling group. The configurations include the labels, taints, and instance types. If a scaling group can simulate the scheduling of pods, this scaling group is selected for the scale-out event. If more than one scaling groups can simulate the scheduling of pods, the system selects the scaling group that has the fewest idle resources left after simulation.

3.1.4.13. Sandboxed-containers

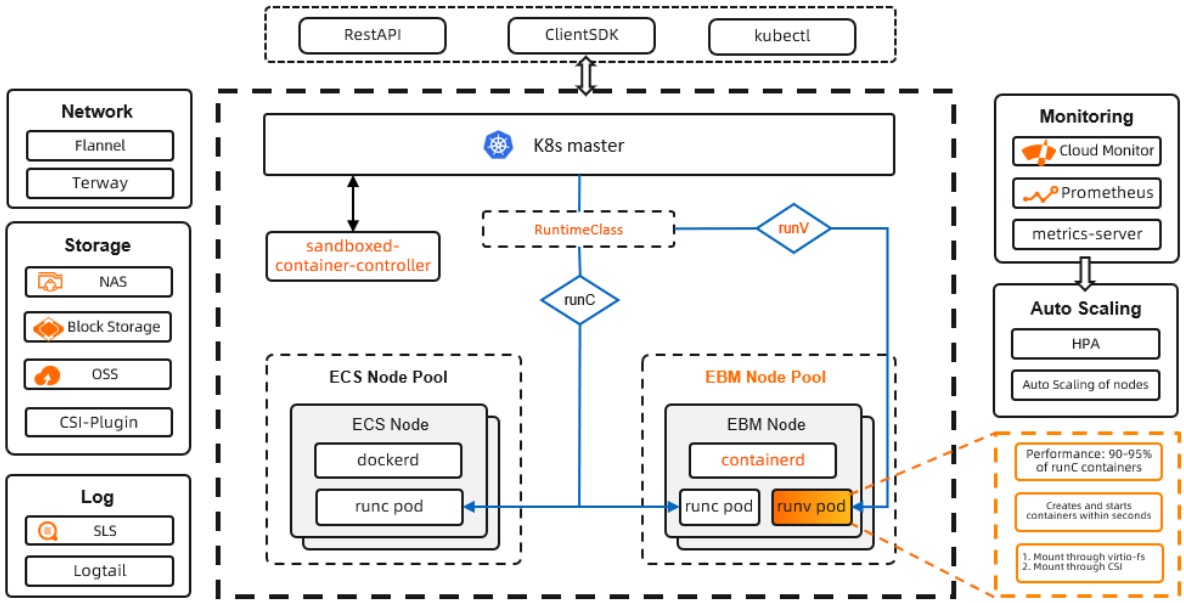
3.1.4.13.1. Overview

Sandboxed-Container is an alternative to the Docker runtime. Sandboxed-Container allows you to run applications in a sandboxed and lightweight virtual machine that has a dedicated kernel. This enhances resource isolation and improves security.

Sandboxed-Container is applicable to scenarios such as untrusted application isolation, fault isolation, performance isolation, and workload isolation among multiple users. Sandboxed-Container provides higher security. Sandboxed-Container has minor impacts on application performance and offers the same user experience as Docker in terms of logging, monitoring, and elastic scaling.



Architecture



Features

Sandboxed-Container is a container-securing runtime that is developed by Alibaba Cloud based on sandboxed and lightweight virtual machines. Compared with Sandboxed-Container V1, Sandboxed-Container V2 maintains the same isolation performance and reduces the pod overhead by 90%. It also allows you to start sandboxed containers 3 times faster and increases the maximum number of pods that can be deployed on a host by 10 times. Sandboxed-Container V2 provides the following key features:

- Strong isolation based on sandboxed and lightweight virtual machines.
- Good compatibility with runC in terms of application management.
- Network Attached Storage (NAS) file systems, disks, and Object Storage Service (OSS) buckets can be mounted both directly and through virtio-fs.
- The same user experience as runC in terms of logging, monitoring, and storage.
- Supports RuntimeClass (runC and runV). For more information, see RuntimeClass.
- Easy to use with minimum requirements on technical skills.
- Higher stability than Kata Containers. For more information about Kata Containers, see Kata Containers.

3.1.4.13.2. Create a Kubernetes cluster that supports sandboxed containers

This topic describes how to create a Kubernetes cluster that runs sandboxed containers in the Container Service console.

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters** and click **Create Kubernetes Cluster** in the upper-right corner.
3. On the **Create Kubernetes Cluster** page, set the parameters.

Parameter	Description

Parameter	Description
Cluster Name	The name of the cluster. The name must be 1 to 63 characters in length, and can contain digits, letters, and hyphens (-). **Note** The cluster name must be unique among clusters that belong to the same Apsara Stack tenant account.
Region	The region where you want to deploy the cluster.
VPC	You can select a VPC from the drop-down list. ○ If the specified VPC is already associated with a NAT gateway, Container Service uses this NAT gateway. ○ Otherwise, the system automatically creates a NAT gateway. If you do not want the system to create a NAT gateway, clear **Configure SNAT for VPC**. **Note** If you disallow the system to automatically create NAT gateways and want the VPC to access the Internet, you must manually associate the VPC with a NAT gateway or create Source Network Address Translation (SNAT) rules for the VPC.
VSwitch	Select one or more vSwitches for the cluster. You can select up to three vSwitches that are deployed in different zones.
Kubernetes Version	Select a Kubernetes version.
Container Runtime	Select a runtime for the Kubernetes cluster.
Billing Method	Only pay-as-you-go nodes are supported.
Master node configurations	Set the Instance Type and System Disk parameters: ○ Master Node Quantity: Three master nodes can be added. ○ Instance Type: You can select one or more instance types. For more information, see *Instance types* in *ECS User Guide*. ○ System Disk: Standard SSDs and ultra disks are supported. **Note** You can select **Enable Backup** to back up disk data.
Worker Instance	By default, **Create Instance** is selected.

Parameter	Description
Worker node configurations	If **Worker Instance** is set to **Create Instance**, set the following parameters: ○ Instance Type: Select Elastic Compute Service (ECS) bare metal instance types. ○ Selected Types: The instance types that you have selected. ○ Quantity: Set the number of worker nodes. ○ System Disk: Standard SSDs and ultra disks are supported. **Note** You can select **Enable Backup** to back up disk data. ○ Mount Data Disk: Standard SSDs and ultra disks are supported. **Note** You can enable disk encryption and data backup when you mount disks. The disks are used to store the root file systems of containers on the nodes. Therefore, you must mount a disk of at least 200 GiB. We recommend that you mount a disk of at least 1 TB.
Password	Set a password that is used to log on to the nodes. **Note** The password must be 8 to 30 characters in length, and must contain at least three of the following types of character: uppercase letters, lowercase letters, digits, and special characters.
Confirm Password	Enter the password again.
Network Plug-in	Flannel and Terway are supported. By default, Flannel is selected.
Pod CIDR Block and Service CIDR	These parameters are optional. For more information, see *Network planning* in *VPC User Guide*. **Note** These parameters are available only when you select an existing VPC.
Configure SNAT	This parameter is optional. If you clear Configure SNAT for VPC, you must create a NAT gateway or configure SNAT rules for the VPC.
Public Access	Specify whether to expose the API server with an elastic IP address (EIP). The Kubernetes API server provides multiple HTTP-based RESTful APIs that can be used to create, delete, modify, query, and watch resource objects such as pods and Services. ○ If you select this check box, an EIP is created and attached to an internal-facing Server Load Balancer (SLB) instance. Port 6443 used by the API server is open on the master nodes. You can connect to and manage the cluster by using kubeconfig over the Internet. ○ If you clear this check box, no EIP is created. You can connect to and manage the cluster only by using kubeconfig from within the VPC.

Parameter	Description
Ingress	Specify whether to install Ingress controllers. **Install Ingress Controller** is selected by default.
Log Service	If you enable Log Service, you can select an existing project or create a project. If you select **Enable Log Service**, the Log Service plug-in is automatically installed in the cluster. If you select **Create Ingress Dashboard**, Ingress access logs are collected and displayed on dashboards.
Volume Plug-in	CSI is selected by default.
Deletion Protection	If you select this check box, the cluster cannot be deleted through the console or API operations.
RDS Whitelist	Add the IP addresses of the nodes to the whitelist of the ApsaraDB RDS instance that is allowed to access the Kubernetes cluster. **Note** To enable an ApsaraDB RDS instance to access the Kubernetes cluster, you must deploy the ApsaraDB RDS instance in the same VPC as the Kubernetes cluster.
Node Protection	This check box is selected by default to prevent nodes from being deleted through the console or API operations.
Labels	Add labels to the cluster.

4. Complete the advanced settings of the cluster.

Parameter	Description
IP Addresses per Node	Set the maximum number of IP addresses that can be assigned to a node.
Kube-proxy Mode	IPVS and iptables are supported. ○ iptables is a kube-proxy mode. It uses iptables rules to conduct service discovery and load balancing. The performance of this mode is restricted by the size of the Kubernetes cluster. This mode is suitable for Kubernetes clusters that run a small number of Services. ○ IPVS is a high-performance kube-proxy mode. It uses Linux Virtual Server (LVS) to conduct service discovery and load balancing. This mode is suitable for Kubernetes clusters that run a large number of Services. We recommend that you use this mode in scenarios where high-performance load balancing is required.
Custom Node Name	Specify whether to use a custom node name.
Node Port Range	Specify the node port range.
Taints	Add taints to all worker nodes in the cluster.
CPU Policy	Set the CPU policy. ○ None: This policy indicates that the default CPU affinity is used. This is the default policy. ○ Static: This policy allows pods with certain resource characteristics on the node to be granted enhanced CPU affinity and exclusivity.

Parameter	Description
Cluster Domain	The default cluster domain name is cluster.local. You can specify a domain name for the cluster.
Cluster CA	Specify whether to enable the cluster CA certificate.
User Data	You can customize the startup behaviors of ECS instances and import data to the ECS instances. The user data can be used to perform the following operations: ○ Run user data scripts upon instance startup. ○ Pass user data as common data into an ECS instance for future reference.

5. Click **Create Cluster** in the upper-right corner of the page.
6. On the **Confirm** page, click **OK**.

Result

After the Kubernetes cluster is created, you can find it on the **Clusters** page in the Container Service console.

3.1.4.13.3. Expand a Container Service cluster that runs sandboxed containers

This topic describes how to scale out the worker nodes in a Container Service cluster that runs sandboxed containers in the Container Service console.

Prerequisites

You cannot scale out the master nodes in a Container Service cluster that runs sandboxed containers.

To expand a Container Service cluster that runs sandboxed containers, you must set the parameters as required in the following table. Otherwise, the added nodes cannot run sandboxed containers.

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters**.
3. On the **Clusters** page, find the cluster that you want to manage and click **Manage** in the **Actions** column.
4. Go to the Expand page and set the required parameters.

   In this example, the number of worker nodes in the Container Service cluster is increased from three to five. The following table describes the required parameters.

Parameter	Description
Cluster Name	By default, the name of the Container Service cluster appears.
Region	The region where the Container Service cluster is deployed.
Container Runtime	By default, Sandboxed-Container appears.

Parameter	Description
VPC	You can select a virtual private cloud (VPC) from the drop-down list. ◦ If the specified VPC has a NAT gateway, Container Service uses this NAT gateway. ◦ Otherwise, the system automatically creates a NAT gateway. If you do not want the system to create a NAT gateway, clear **Configure SNAT for VPC**. **Note** If the system does not automatically create a NAT gateway, you must manually configure a NAT gateway to ensure secure communication between the VPC and the Internet. You can also manually configure the Source Network Address Translation (SNAT) feature.
VSwitch	Select one or more vSwitches for the Container Service cluster. You can select up to three vSwitches that are deployed in different zones.
Billing Method	Only pay-as-you-go nodes are supported.
Existing Worker Nodes	The number of existing workers in the Container Service cluster.
Nodes to Add	Set the number of worker nodes to add.
Worker Nodes After Scaling	The number of workers after the scaling.
Instance Type	Select ECS Bare Metal Instance.
Selected Types	The selected instance types.
System Disk	Standard SSDs and ultra disks are supported. **Note** You can select **Enable Backup** to back up disk data.
Mount Data Disk	Standard SSDs and ultra disks are supported. **Note** When you mount data disks, you can enable disk encryption and data backup. Data disks are used to store the root file systems of containers on the nodes. Therefore, you must mount a data disk of at least 200 GiB. We recommend that you mount a data disk of at least 1 TB.
Password	◦ **Password**: Enter the password that is used to log on to the nodes. ◦ **Confirm Password**: Enter the password again.
RDS Whitelist	Set the RDS whitelist. Add the IP addresses of the scaled nodes to the RDS whitelist.
Label	Add labels to the nodes.
Taint	Add taints to worker nodes in the Container Service cluster.

Parameter	Description
CPU Policy	Set the CPU policy. ○ none: the default CPU policy. This policy enables the default CPU affinity scheme. ○ static: This policy allows pods with specific resource characteristics on the node to be configured with enhanced CPU affinity and exclusivity.
User Data	Customize the startup behaviors of Elastic Compute Service (ECS) instances and import data to the ECS instances. The user data can be used to perform the following operations: ○ Run user data scripts during instance startup. ○ Pass user data as common data into an ECS instance for future reference.

5. Click **Submit**.

What's next

After the Container Service cluster is expanded, go to the details page of the Container Service cluster. In the left-side navigation pane, choose **Clusters > Node Pools**. You can find that the number of worker nodes is increased from 3 to 5.

3.1.4.13.4. Create an application that runs in sandboxed containers

This topic describes how to use an image to create an NGINX application that runs in sandboxed containers. The NGINX application is accessible over the Internet.

Prerequisites

A cluster that contains sandboxed containers is created. For more information, see Create a Kubernetes cluster that supports sandboxed containers.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Applications > Deployments**.
3. In the upper-left corner of the **Deployments** page, select the cluster and namespace.
4. In the upper-right corner of the **Deployments** page, click **Create from Image**.
5. On the **Basic Information** wizard page, set the basic information and parameters for the application and click **Next**.

   Set **Container Runtime** to **runv**. Set the following parameters: **Name**, **Replicas**, **Type**, **Label**, and **Annotations**. Select whether you want to enable **Synchronize Timezone**. The number of replicas specifies the number of pods that are provisioned for in the application.

   > **Note**
   >
   > In this example, **Deployment** is selected as the application type.

6. Configure the containers.

> **Note** At the top of the **Container** wizard page, click **Add Container** to add more containers for the application.

The following table describes the parameters that are required to configure the containers.

- General settings

Parameter	Description
Image Name	Click **Select Image**. In the dialog box that appears, select an image and click **OK**. In this example, an NGINX image is selected. You can also enter the address of a private registry. The registry address must be in the following format: `domainname/namespace/imagename:tag`.
Image Version	■ Click **Select Image Version** and select an image version. If you do not specify an image version, the latest image version is used. ■ You can select the following image pull policies: ■ **ifNotPresent**: If the image that you want to pull is found in the region where the cluster is deployed, the local image is used. Otherwise, Container Service pulls the image from the corresponding repository. ■ **Always**: Container Service pulls the image from the repository each time the application is deployed or expanded. ■ **Never**: Container Service uses only local images. **Note** If you select **Image Pull Policy**, no image pull policy is applied to the Deployment of the application. ■ To pull the image without a Secret, click **Set Image Pull Secret** to set a Secret for pulling images.
Resource Limit	You can specify an upper limit for the CPU, memory, and ephemeral storage space that the container can consume. This prevents the container from occupying an excessive amount of resources. The CPU resource is measured in milicores (one thousandth of one core). The memory resource is measured in MiB. The ephemeral storage resource is measured in GiB.
Required Resources	The amount of CPU and memory resources that are reserved for this application. These resources are exclusive to the container. This prevents the application from becoming unavailable when other services or processes occupy these resources.
Container Start Parameter	■ stdin: specifies that start parameters defined in the console are sent to the container. ■ tty: specifies that start parameters defined in a virtual terminal are sent to the console.
Privileged Container	■ If you select Privileged Container, privileged=true is set for the container and the privilege mode is enabled. ■ If you do not select Privileged Container, privileged=false is set for the container and the privilege mode is disabled.

Parameter	Description
Init Container	If you select Init Container, an init container is created. An init container provides tools for managing pods. For more information, see Init Containers.

- (Optional)Ports

  Specify the container ports.

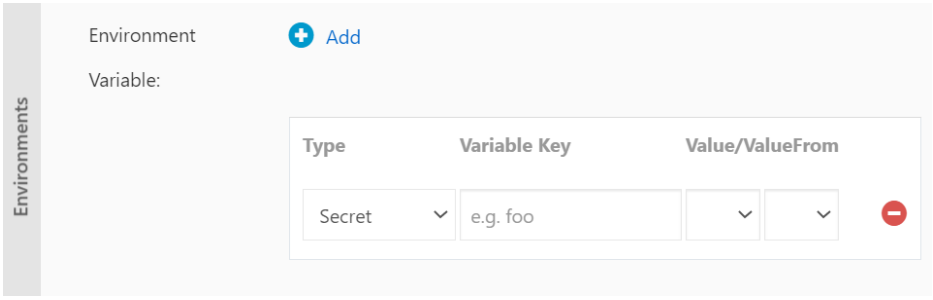
  - Name: Enter a name for the port.
  - Container Port: Enter the container port that you want to open. Enter a port number from 1 to 65535.
  - Protocol: Select TCP or UDP.

- (Optional)Environments

  You can use key-value pairs to set environment variables for pods. Environment variables are used to apply pod configurations to containers. For more information, see Pod variables.

  - Type: Select the type of the environment variable. You can select **Custom**, **ConfigMaps**, **Secret**, **Value/ValueFrom**, or **ResourceFieldRef**. If you select ConfigMaps or Secret as the type of the environment variable, all values in the selected ConfigMap or Secret are passed to the container environment variables. In this example, Secret is selected.

    Select **Secret** from the Type drop-down list and select a Secret from the **Value/ValueFrom** drop-down list. All values in the selected Secret are passed to the environment variable.

    In this case, the YAML file that is used to deploy the application contains the settings that reference all values in the specified Secret.

    ```
    envFrom:
      - secretRef:
          name: test
    ```

  - Variable Key: Specify the key of the environment variable.
  - Value/ValueFrom: This reference is used to define the environment variable.

- (Optional)Health Check

  Health check settings include liveness and readiness probes. Liveness probes determine when to restart the container. Readiness probes indicate whether the container is ready to accept network traffic. For more information about health checks, see Configure Liveness, Readiness, and Startup Probes.

Request type	Parameter

Request type	Parameter
HTTP request	Sends an HTTP GET request to the container. You can set the following parameters: ■ Protocol: Select HTTP or HTTPS. ■ Path: Enter the requested path on the server. ■ Port: Enter the container port that you want to open. Enter a port number from 1 to 65535. ■ HTTP Header: Enter the custom headers in the HTTP request. Duplicate headers are allowed. Key-value pairs are supported. ■ Initial Delay (s): the initialDelaySeconds field in the YAML file. This field specifies the wait time (in seconds) before the first probe is performed after the container is started. Default value: 3. ■ Period (s): the periodSeconds field in the YAML file. This field specifies the frequency (in seconds) that the probe is performed. Default value: 10. Minimum value: 1. ■ Timeout (s): the timeoutSeconds field in the YAML file. This field specifies the time (in seconds) after which the probe times out. Default value: 1. Minimum value: 1. ■ Healthy Threshold: Enter the minimum number of consecutive successes that must occur before a container is considered healthy after a failed probe. Default value: 1. Minimum value: 1. For liveness probes, this parameter must be set to 1. ■ Unhealthy Threshold: Enter the minimum number of consecutive failures that must occur before a container is considered unhealthy after a success. Default value: 3. Minimum value: 1.
TCP connection	Sends a TCP socket to the container. Kubelet attempts to open the socket on the specified port. If the connection can be established, the container is considered healthy. Otherwise, the container is considered unhealthy. You can set the following parameters: ■ Port: Enter the container port that you want to open. Enter a port number from 1 to 65535. ■ Initial Delay (s): the initialDelaySeconds field in the YAML file. This field specifies the wait time (in seconds) before the first probe is performed after the container is started. Default value: 15. ■ Period (s): the periodSeconds field in the YAML file. This field specifies the frequency (in seconds) that the probe is performed. Default value: 10. Minimum value: 1. ■ Timeout (s): the timeoutSeconds field in the YAML file. This field specifies the time (in seconds) after which the probe times out. Default value: 1. Minimum value: 1. ■ Healthy Threshold: Enter the minimum number of consecutive successes that must occur before a container is considered healthy after a failed probe. Default value: 1. Minimum value: 1. For liveness probes, this parameter must be set to 1. ■ Unhealthy Threshold: Enter the minimum number of consecutive failures that must occur before a container is considered unhealthy after a success. Default value: 3. Minimum value: 1.

Request type	Parameter
Command	Runs a probe command in the container to check the health status of the container. You can set the following parameters: ■ Command: Enter the probe command that is run to check the health status of the container. ■ Initial Delay (s): the initialDelaySeconds field in the YAML file. This field specifies the time (in seconds) to wait before the first probe is performed after the container is started. Default value: 5. ■ Period (s): the periodSeconds field in the YAML file. This field specifies the frequency (in seconds) that the probe is performed. Default value: 10. Minimum value: 1. ■ Timeout (s): the timeoutSeconds field in the YAML file. This field specifies the time (in seconds) after which the probe times out. Default value: 1. Minimum value: 1. ■ Healthy Threshold: Enter the minimum number of consecutive successes that must occur before a container is considered healthy after a failed probe. Default value: 1. Minimum value: 1. For liveness probes, this parameter must be set to 1. ■ Unhealthy Threshold: Enter the minimum number of consecutive failures that must occur before a container is considered unhealthy after a success. Default value: 3. Minimum value: 1.

- Lifecycle

  You can set the following parameters to configure the lifecycle of the container: Start, Post Start, and Pre Stop. For more information, see Configure the lifecycle of a container.

  - **Start**: Set the command and parameter that take effect before the container starts.
  - **Post Start**: Set the command that takes effect after the container starts.
  - **Pre Stop**: Set the command that takes effect before the container stops.

- (Optional)Volume

  You can mount local storage volumes and persistent volume claims (PVCs) to the container.

  - **Local Storage**: You can select HostPath, ConfigMap, Secret, and EmptyDir. The source directory or file is mounted to a path in the container. For more information, see Volumes.
  - **PVC**: Select Cloud Storage.

  In this example, a PVC named disk-ssd is mounted to the */tmp* path of the container.

- (Optional)Log

  Configure **Log Service**. You can specify collection configurations and custom tags.

  > **Notice** Make sure that the Log Service agent has been installed for the cluster.

Parameter	Description
	Logstore: creates a Logstore in Log Service to store collected logs.

Parameter Collection configuration	Description
	Log Path in Container: specifies stdout or a path to collect logs. ■ stdout: specifies that the stdout files are collected. ■ Text Logs: specifies that logs in the specified path of the container are collected. In this example, */var/log/nginx* is specified as the path. Wildcard characters can be used to specify the path.
Custom Tag	You can also set custom tags. Custom tags are added to the logs of the container when the logs are collected. Custom tags provide an easy method to filter collected logs and perform statistical analytics.

7. Set the preceding parameters based on your business requirements and click **Next**.
8. (Optional)Configure advanced settings.
   - Access Control

> **Note**
>
> You can configure the following access control settings based on your business requirements:
>
> - Internal applications: For applications that run inside the cluster, you can create a service of the ClusterIP or NodePort type to enable internal communication.
> - External applications: For applications that are exposed to the Internet, you can configure access control by using one of the following methods:
>   - Create a service of the LoadBalancer type and enable access to your application over the Internet by using a Server Load Balancer (SLB) instance.
>   - Create an Ingress to route external access to a service inside the cluster. For more information, see Ingress.

Configure the access control settings to enable access to pods that run the application. In this example, a ClusterIP service and an Ingress are created to enable access to the NGINX application over the Internet.

Parameter	Description
Services	Click **Create** on the right side of **Service**. In the Create Service dialog box, set the parameters. Select **Cluster IP**.
Ingress	Click **Create** on the right side of **Ingresses**. In the Create dialog box, set the parameters. **Note** When you deploy an application from an image, you can create an Ingress for only one service. In this example, a virtual hostname is specified as the test domain name. You must add a mapping rule for this domain name to the hosts file, as shown in the following code block. In actual scenarios, use a domain name that has obtained an Internet Content Provider (ICP) number. `101.37.224.146  foo.bar.com  #The IP address of the Ingress.`

You can find the created service and Ingress in the **Access Control** section. Click **Update** or **Delete** to modify the settings.

- Scaling

  Specify whether to enable **HPA** to automatically scale the number of pods based on the CPU and memory usage. This enables the application to run smoothly at different load levels.

  > **Note** To enable Horizontal Pod Autoscaler (HPA), you must configure resources that support scaling for the container. Otherwise, HPA does not take effect.

  - **Metric**: Select CPU Usage or Memory Usage. The selected resource type must be the same as the one you have specified in the Required Resources field.
  - **Condition**: Specify the resource usage threshold. HPA triggers scaling events when the threshold is exceeded.
  - **Max. Replicas**: Specify the maximum number of replicated pods to which the application can be scaled.
  - **Min. Replicas**: Specify the minimum number of replicated pods that must run.

- Scheduling

  You can set the following parameters: Update Method, Node Affinity, Pod Affinity, Pod Anti Affinity, and Toleration. For more information, see Affinity and anti-affinity.

  > **Note** Node affinity and pod affinity affect pod scheduling based on node labels and pod labels. You can add node labels and pod labels that are provided by Kubernetes to configure node affinity and pod affinity. You can also add custom labels to nodes and pods, and then configure node affinity and pod affinity based on these custom labels.

Parameter	Description
Update Method	Select Rolling Update or OnDelete. For more information, see Deployments.
Node Affinity	Add labels to worker nodes to set **Node Affinity**. Node Affinity supports required and preferred rules, and various operators, such as In, NotIn, Exists, DoesNotExist, Gt, and Lt. ■ **Required**: Specify node labels that must be matched for pod scheduling. In the YAML file, these rules are defined by the requiredDuringSchedulingIgnoredDuringExecution field of the nodeAffinity parameter. These rules have the same effect as the `nodeSelector` parameter. In this example, pods can be scheduled to only nodes with the specified labels. You can create multiple required rules. However, only one of them must be met. ■ **Preferred**: Specify the node weight and node labels that are not required to be matched for pod scheduling. Pods are scheduled to a node that matches the preferred rules when multiple nodes match the required rules. In the YAML file, these rules are defined by the preferredDuringSchedulingIgnoredDuringExecution field of the nodeAffinity parameter. In this example, the scheduler attempts to schedule the pod to a node that matches the preferred rules. You can set node weights in preferred rules. If multiple nodes match the required and preferred rules, the node with the highest weight is preferred for pod scheduling. You can create multiple preferred rules. However, all of them must be met before the pod can be scheduled.

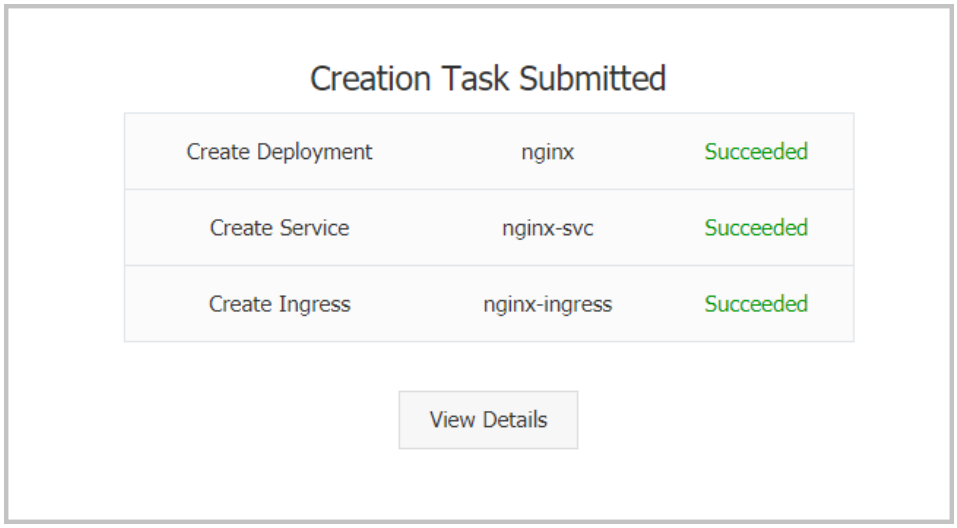
Parameter	Description
Pod Affinity	Pod affinity specifies that pods can be scheduled to nodes or topological domains where pods with matching labels are deployed. For example, you can use pod affinity to deploy services that communicate with each other to the same topological domain, such as a host. This reduces the network latency between these services. Pod affinity enables you to specify to which node pods can be scheduled based on the labels on other running pods. Pod affinity supports required and preferred rules, and the following operators: `In`, `NotIn`, `Exists`, and `DoesNotExist`. ■ **Required**: Specify rules that must be matched for pod scheduling. In the YAML file, these rules are defined by the requiredDuringSchedulingIgnoredDuringExecution field of the podAffinity parameter. A node must match the required rules before pods can be scheduled to the node. ■ **Namespace**: Specify the namespace to apply the required rule. Pod affinity rules are defined based on the labels that are added to pods and therefore must be scoped to a namespace. ■ **Topological Domain**: Set the topologyKey. This specifies the key for the node label that the system uses to denote the topological domain. For example, if you set the parameter to `kubernetes.io/hostname`, topologies are determined by nodes. If you set the parameter to `beta.kubernetes.io/os`, topologies are determined by the operating systems of nodes. ■ **Selector**: Click Add to add pod labels. ■ **View Applications**: Click **View Applications** and set the namespace and application in the dialog box that appears. You can view the pod labels on the selected application and add the labels as selectors. ■ Required Rules: Specify labels on existing applications, the operator, and the label value. In this example, the required rule specifies that the application to be created is scheduled to a host that runs applications with the `app:nginx` label. ■ **Preferred**: Specify rules that are not required to be matched for pod scheduling. In the YAML file, preferred rules are defined by the preferredDuringSchedulingIgnoredDuringExecution field of the podAffinity parameter. The scheduler attempts to schedule the pod to a node that matches the preferred rules. You can set node weights in preferred rules. Set the other parameters as described in the preceding settings. **Note** **Weight**: Set the weight of a preferred rule to a value from 1 to 100. The scheduler calculates the weight of each node that meets the preferred rule based on an algorithm, and then schedules the pod to the node with the highest weight.

Parameter	Description
Pod Anti Affinity	Pod anti-affinity rules specify that pods are not scheduled to topological domains where pods with matching labels are deployed. Pod anti-affinity rules apply to the following scenarios: ■ Schedule the pods of an application to different topological domains, such as multiple hosts. This allows you to enhance the stability of the service. ■ Grant a pod exclusive access to a node. This enables resource isolation and ensures that no other pod can share the resources of the specified node. ■ Schedule pods of an application to different hosts if the pods may interfere with each other. **Note** The parameters of pod anti-affinity rules are the same as those of pod affinity rules. You can create the rules for different scenarios.
Toleration	Set toleration rules to allow pods to be scheduled to nodes with matching taints.
Schedule to Virtual Nodes	Specify whether to schedule pods to virtual nodes. This option is unavailable if the cluster does not contain a virtual node.

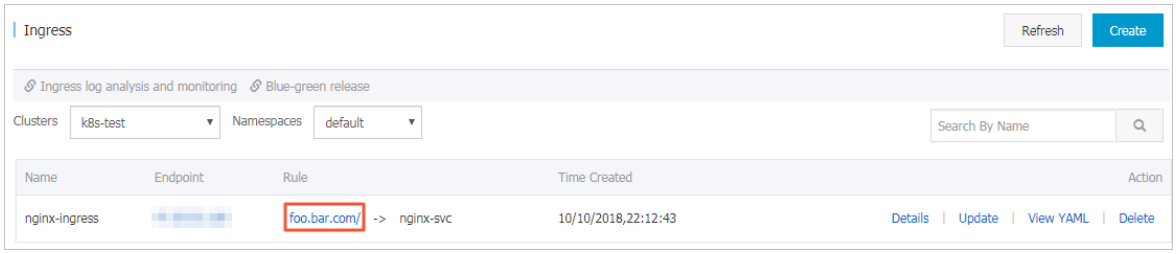
- Labels and Annotations
  - Pod Labels: Add a label to the pod. The label is used to identify the application.
  - Pod Annotations: Add an annotation to the pod.

9. Click **Create**.

   After the application is deployed, you are redirected to the Complete page. You can find the resource objects under the application and click **View Details** to view application details.



10. In the left-side navigation pane, choose **Ingresses and Load Balancing > Ingresses**.
    The created Ingress rule is displayed on the page.



Result

Enter the test domain name into the address bar of your browser and press the Enter key. The NGINX welcome page appears.

3.1.4.13.5. Configure a Kubernetes cluster that runs both sandboxed and Docker containers

Node pools support multiple types of container runtime. However, nodes in the same node pool must use the same type of container runtime. Container Service allows you to create node pools of different container runtime types for a cluster. This topic describes how to create a node pool of sandboxed containers and a node pool of Docker containers for a Kubernetes cluster.

Prerequisites

A Kubernetes cluster is created in the Container Service console. For more information, see Create a Kubernetes cluster.

> **Notice** The Kubernetes cluster must meet the following requirements:
>
> - The Kubernetes version must be 1.14.6-aliyun.1 or later.
> - The network plug-in must be Flannel or Terway. Terway must run in One ENI for Multi-Pod mode.
> - The volume plug-in must be CSI 1.14.8.39-0d749258-aliyun or later. Flexvolume is not supported.
> - The Logtail version must be 0.16.34.2-f6647154-aliyun or later.

Usage notes

- By default, you can deploy up to 100 nodes in a Kubernetes cluster. To increase the quota, click here to submit a ticket.
- Before you add an existing Elastic Compute Service (ECS) instance that is deployed in a virtual private cloud (VPC), make sure that an elastic IP address (EIP) is attached to the ECS instance, or a Network Address Translation (NAT) gateway is created in the VPC. In addition, the nodes that you want to add to the node pool must have Internet access. Otherwise, the ECS instance cannot be added.

Create a node pool that runs Docker containers

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Nodes**.
3. On the **Node Pools** page, click **Create Node Pool** and set the parameters.

   For more information, see Create a Kubernetes cluster. The following table describes the parameters.

Parameter	Description
Name	Specify a name for the node pool.
Container Runtime	Select Docker. This specifies that all containers in the node pool are Docker containers.
Quantity	Specify the initial number of nodes in the node pool. If you do not need to create nodes, select 0.
Operating System	The CentOS and Aliyun Linux operating systems are supported.
ECS Label	You can add labels to the ECS instances.
Node Label	You can add labels to the nodes in the cluster.
Custom Resource Groups	Specify the resource groups of nodes to be scaled out in the node pool.
Custom Security Group	Configure the security group for the cluster.

4. Click **OK**.

Create a node pool that runs sandboxed containers

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Nodes**.
3. On the **Node Pools** page, click **Create Node Pool** and set the parameters.

   For more information, see Create a Kubernetes cluster. The following table describes the parameters.

Parameter	Description
Name	Specify a name for the node pool.
Container Runtime	Select Sandboxed-Container. This specifies that all containers in the node pool are sandboxed containers.
Quantity	Specify the initial number of nodes in the node pool. If you do not need to create nodes, select 0.
Operating System	Select an operating system. Sandboxed containers support only the Aliyun Linux operating system.
Mount Data Disk	You must mount a disk of at least 200 GiB.
ECS Label	You can add labels to the ECS instances.
Node Label	You can add labels to the nodes in the cluster.
Custom Resource Groups	Specify the resource groups of nodes to be scaled out in the node pool.

Parameter	Description
Custom Security Group	Configure the security group for the cluster.

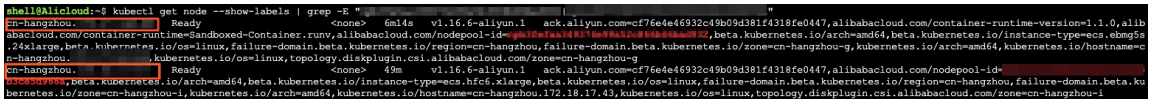
4. Click **OK**.

Result

- After you perform the preceding steps, check the status of the node pool on the **Node Pools** page. If the node pool is in the **Activated** state, the node pool is created.
- To view detailed information about the nodes in the node pool, connect to the Kubernetes cluster where the node pool is deployed.
  i. On the **Node Pools** page, click the name of the node pool that you have created. In the **Node Pool Information** section, find and record the **Node Pool ID**.

  ii. For more information about how to connect to a Kubernetes cluster, see Connect to a cluster through kubectl.
  iii. Run the following command to query the name of a specified node:

```
kubectl get node --show-labels | grep -E "${node pool ID}|${node pool ID}"
```



  iv. Run the following command to query detailed information about a specified node:

```
kubectl get node -o wide | grep -E "${node name} {node name}"
```

3.1.4.13.6. How do I select between Docker and Sandboxed-Container?

Containers and images have become the industry standards for software packaging and delivery. Kubernetes has become a standard platform for building, developing, and managing containerized cloud-native applications. An increasing number of enterprises and customers choose to deploy their applications in Container Service. Container Service supports two types of runtime: Docker and Sandboxed-Container. This topic describes the differences between these runtimes in the following aspects: implementations and limits, commonly used commands provided by Docker Engine and Containerd, and deployment architectures. This provides references for you to select between Docker and Sandboxed-Container based on your requirements.

Differences between Docker and Sandboxed-Container in terms of implementations and limits

Item	Docker	Sandboxed-Container V2	Description
Cluster type	All types	All types	N/A
Node type	• ECS • EBM	EBM	N/A

Item	Docker	Sandboxed-Container V2	Description
Node operating system	• CentOS • Alibaba Cloud Linux2	Alibaba Cloud Linux2	• You cannot deploy both Docker and Sandboxed-Container on the same node. • To deploy both Docker and Sandboxed-Container in a cluster, you can create node pools of different runtime types.
Container engine	Docker	Containerd	N/A
Monitoring	Supported	Supported	N/A
Container log collection	Supported	Sidecar: supported. Manual configuration is required.	N/A
Container stdout collection	Supported	Supported	N/A
RuntimeClass	Not supported	Supported (runV)	N/A
Pod scheduling	No configuration is required.	• For Kubernetes V1.14.x, you must add the following configuration to the nodeSelector field: `alibabacloud.com/sandboxed-container: Sandboxed-Container.runv` • For Kubernetes V1.16.x and later, no configuration is required.	N/A
HostNetwork	Supported	Not supported	N/A
exec/logs	Supported	Supported	N/A
Node data disk	N/A	Required. The data disk must be at least 200 GiB.	N/A
Network plug-in	• Flannel • Terway	• Flannel • Terway: supports only the One ENI for Multi-Pod mode.	N/A
Kube-proxy mode	• Iptables • IPVS	• Iptables • IPVS	N/A

Item	Docker	Sandboxed-Container V2	Description
Volume plug-in	CSI Plugin	CSI Plugin	N/A
Container root file system	OverlayFS	VirtioFS	N/A

Differences in the commonly used commands provided by Docker Engine and Containerd

Docker uses Docker Engine for container lifecycle management. Sandboxed-Container uses Containerd for container lifecycle management. These tools support different commands that can be used to manage images and containers. The following table lists the commonly used commands.

Command	Docker	Containerd	
	docker	crictl (recommended)	ctr
Queries containers	docker ps	crictl ps	ctr -n k8s.io c ls
Queries container details	docker inspect	crictl inspect	ctr -n k8s.io c info
Queries container logs	docker logs	crictl logs	N/A
Runs a command in a container	docker exec	crictl exec	N/A
Mounts local standard input, output, and error streams to a running container	docker attach	crictl attach	N/A
Queries resource usage statistics	docker stats	crictl stats	N/A
Creates a container	docker create	crictl create	ctr -n k8s.io c create
Starts one or more containers	docker start	crictl start	ctr -n k8s.io run
Stops one or more containers	docker stop	crictl stop	N/A
Removes one or more containers	docker rm	crictl rm	ctr -n k8s.io c del
Queries images	docker images	crictl images	ctr -n k8s.io i ls
Queries image details	docker inspect	crictl inspecti	N/A
Pulls an image	docker pull	crictl pull	ctr -n k8s.io i pull
Pushes an image	docker push	N/A	ctr -n k8s.io i push
Removes one or more images	docker rmi	crictl rmi	ctr -n k8s.io i rm
Queries pods	N/A	crictl pods	N/A
Queries pod details	N/A	crictl inspectp	N/A

Command	Docker	Containerd	
	docker	crictl (recommended)	ctr
Starts one or more pods	N/A	crictl runp	N/A
Stops one or more pods	N/A	crictl stopp	N/A

Differences between Docker and Sandboxed-Container in terms of deployment architectures

Runtime	Deployment architecture
Docker	`kubelet -> dockerd -> containerd -> containerd-shim -> runC containers`

Sandboxed-Container V1	`kubelet -> (CRI)containerd`
	`\-> containerd-shim -> runC containers`
	`\-> containerd-shim-kata-v2 -> runV sandboxed containers`

Sandboxed-Container V2	`kubelet -> (CRI)containerd`
	`\-> containerd-shim -> runC containers`
	`\-> containerd-shim-rund-v2 -> runV sandboxed containers`

3.1.4.13.7. Benefits of Sandboxed-Container

This topic describes the advantages and application scenarios of Sandboxed-Container and provides a comparison between Sandboxed-Container and open source Kata Containers. This allows you to learn more about the benefits of Sandboxed-Container.

Context

Sandboxed-Container provides an alternative to the Docker runtime environment. It supports the following features:

- Sandboxed-Container allows your applications to run in a sandboxed and lightweight virtual machine. This virtual machine is equipped with a dedicated kernel and provides better isolation and enhanced security.
- Compared with open source Kata Containers, Sandboxed-Container is optimized for storage, networking, and stability.
- You can use Sandboxed-Container to isolate untrusted applications and applications of different tenants for higher security. You can also use Sandboxed-Container to isolate applications with faults and applications with degraded performance. This minimizes the negative impact on your service. In addition, Sandboxed-Container offers the same user experience as Docker in terms of logging, monitoring, and elastic scaling.

Benefits

Compared with Docker, Sandboxed-Container has the following benefits:

- Strong isolation based on sandboxed and lightweight virtual machines.
- Compatibility with runC in terms of application management.

- High performance that corresponds to 90% performance of applications based on runC.
- The same user experience as runC in terms of logging, monitoring, and storage.
- Support for RuntimeClass.
- Easy to use with limited expertise that is required to use virtual machines.
- Higher stability than that provided by Kata Containers.

Comparison between Sandboxed-Container and Kata Containers

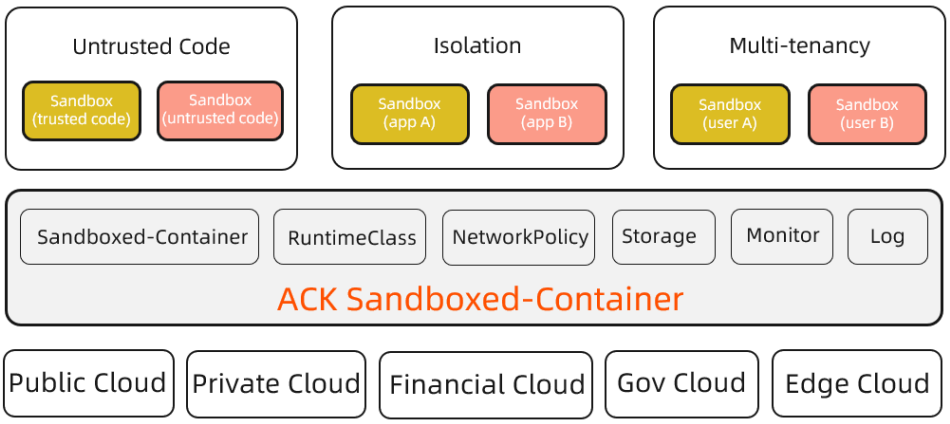
Sandboxed-Container outperforms Kata Containers in the following aspects.

Item	Category	Sandboxed-Container	Kata Containers
Sandbox startup time consumption		About 150 ms	About 500 ms
Root file system		OverlayFS over virtio-fs. Performance: ☆☆☆☆	OverlayFS over 9pfs. Performance: ☆☆
Volume	HostPath	Disks are mounted to Sandboxed-Container over 9pfs. Performance: ☆☆	Disks are mounted to Kata Containers over 9pfs. Performance: ☆☆
	EmptyDir	over VirtioFS	By default, the volume is mounted to Kata Containers over 9pfs.
	Disk	By default, cloud disks are mounted to Sandboxed-Container over virtio-fs. Performance: ☆☆☆☆	Cloud disks are mounted to Kata Containers over 9pfs. Performance: ☆☆
	NAS	By default, Apsara File Storage NAS (NAS) file systems are mounted to Sandboxed-Container over virtio-fs. Performance: ☆☆☆☆	NAS file systems are mounted to Kata Containers over 9pfs. Performance: ☆
Network plug-in		• The Terway network plug-in is used. Its network performance is 20% to 30% higher than Flannel. Terway supports features such as NetworkPolicy. This allows you to define the networking policies for pods. • Flannel	Flannel

Item	Category	Sandboxed-Container	Kata Containers
Monitoring and alerting		• Enhanced monitoring of disks and network conditions for pods that host Sandboxed-Container. • Integrated with Cloud Monitor. This facilitates cluster monitoring and alerting.	Monitoring of disks and network conditions is unavailable for pods that host Sandboxed-Container.
Stability		☆☆☆☆☆	☆☆

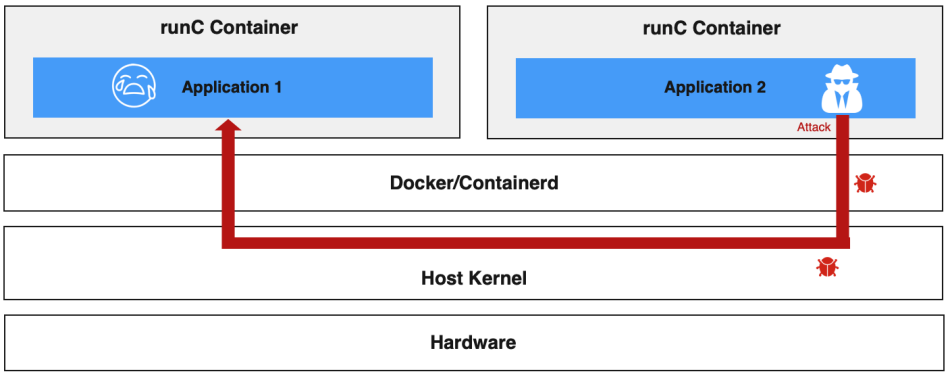
Applicable scenarios of Sandboxed-Container

This section describes the applicable scenarios of Sandboxed-Container.



- Scenario 1: Sandboxed-Container can run untrusted code and applications in isolated containers. This is not supported by containers in runC.

- Security risks of runC



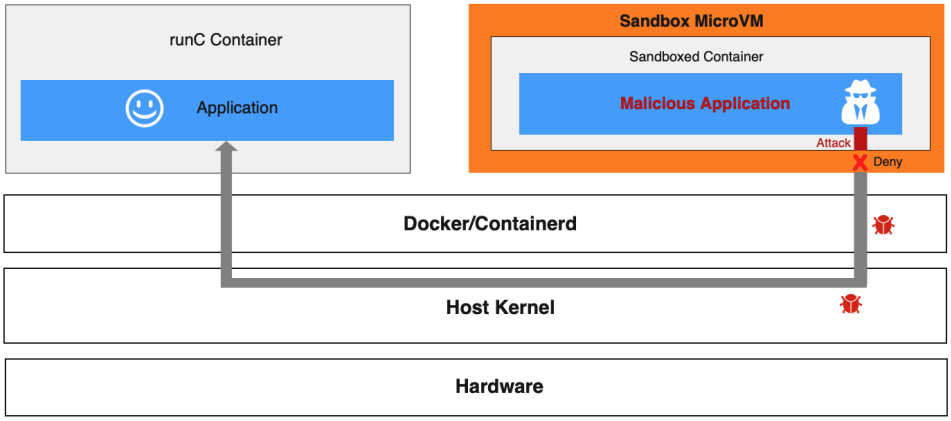
  - runC isolates containers by using namespaces and control groups (cgroups). This exposes containers to security threats.
  - All containers on a node share the host kernel. If a kernel vulnerability is exposed, malicious code may escape to the host and then infiltrate the backend network. Malicious code execution may cause privilege escalation, compromise sensitive data, and destroy system services and other applications.
  - Attackers may also exploit application vulnerabilities to infiltrate the internal network.

  You can implement the following measures to reduce security risks of containers in runC.

  - Seccomp: filters system calls.
  - SElinux: restricts the permissions of container processes, files, and users.
  - Capability: limits the capability of container processes.
  - dockerd rootless mode: forbids users to use root permissions to run the Docker daemon and containers.

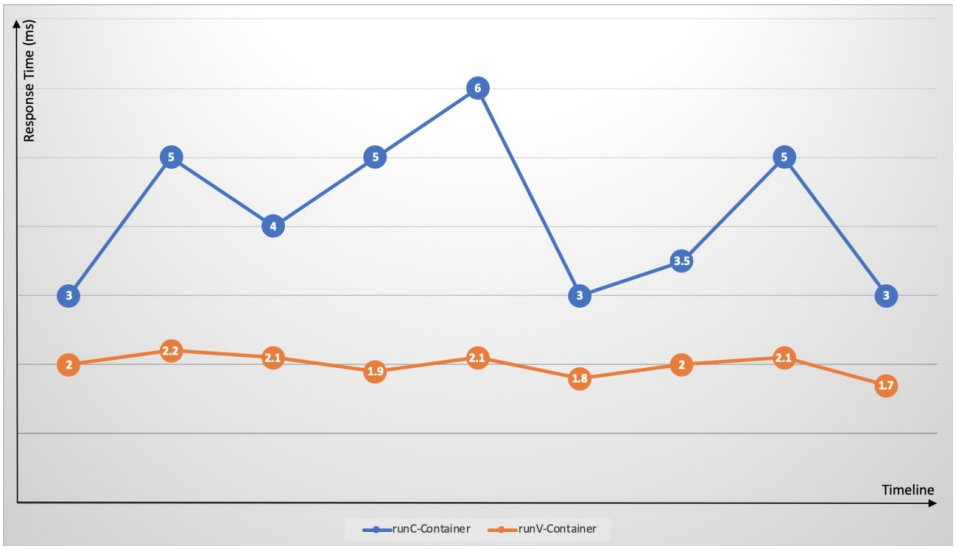
  The preceding measures can enhance the security of containers in runC and reduce attacks on the host kernel by malicious containers. However, container escapes and host kernel vulnerabilities remain unresolved.

- Sandboxed-Container prevents potential risks based on container isolation



  In a Sandboxed-Container runtime environment, applications that have potential security risks are deployed on sandboxed and lightweight virtual machines. Each of the virtual machines has a dedicated guest OS kernel. If a security vulnerability is detected on a guest OS kernel, the attack is limited to one sandbox and does not affect the host kernel or other containers. The Terway network plug-in allows you to define networking policies for pods. This enables system isolation, data isolation, and network isolation for Sandboxed-Containers.

- Scenario 2: Sandboxed-Container resolves common issues of runC containers, such as fault spreading, resource contention, and performance interference.

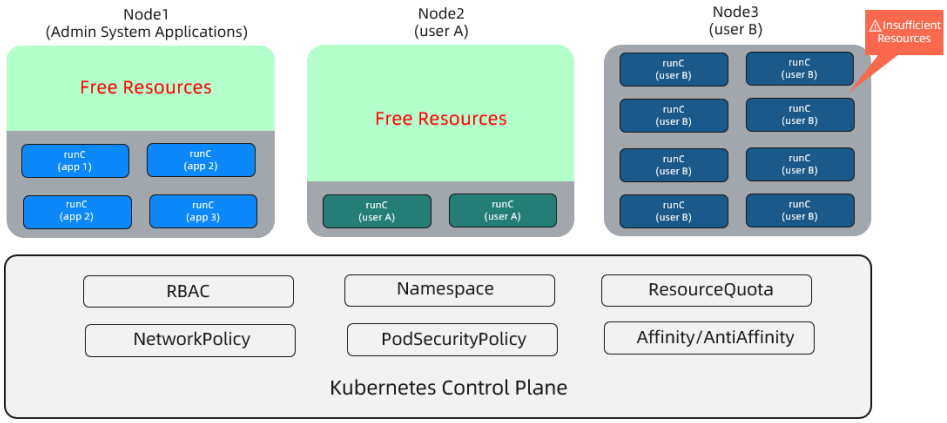


Kubernetes provides easy deployment of different containers on a single node. However, cgroups are not optimized to address resource contention. Resource-intensive applications (such as CPU-intensive, I/O-intensive applications) may compete for the same resources. This causes significant fluctuations in response time and increases the overall response time. Exceptions or faults on an application may spread to the hosting node and disrupt the running of the total cluster. For example, memory leaks and frequent core dumps of an application may overload the node, and exceptions on a container may trigger a host kernel bug that results in complete system failure. Sandboxed-Container addresses the issues that are common with runC containers by using dedicated guest OS kernels and hypervisors. The issues include failure spreading, resource contention, and performance interference.

- Scenario 3: Sandboxed-Container supports multi-tenant services.

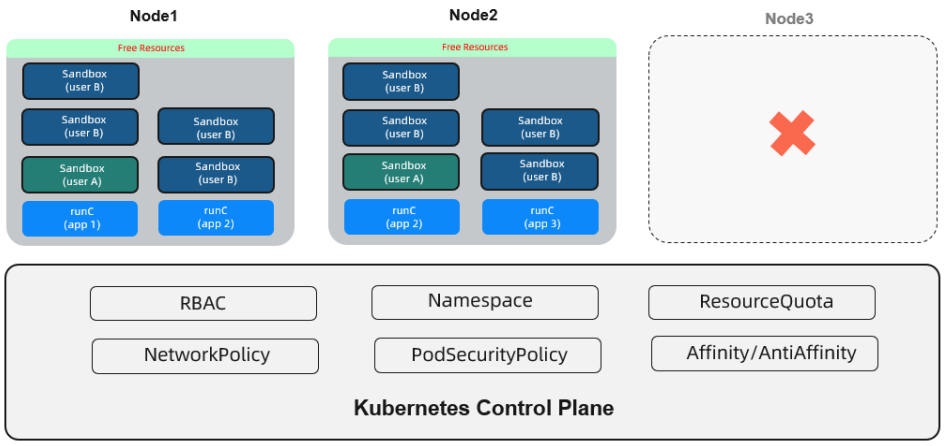
  You may need to isolate the applications of an enterprise that consists of multiple business lines or departments. For example, a financial department requires high security applications. However, other non-security-sensitive applications do not have high security requirements. Containers in runC fail to eliminate the potential risks that arise in untrusted applications. In this scenario, you can implement the following counter measures:

  - Deploy multiple independent single-tenant clusters. For example, deploy financial business and other non-security-sensitive business in different clusters.
  - Deploy a multi-tenant cluster and separate applications of different business lines by namespaces. The resource of a node is exclusive to a single business line. This solution provides data isolation for coordination with the resource quotas and network policies to implement multi-tenant isolation. Compared with multiple single-tenant clusters, this solution focuses on fewer management planes and thus reduces management costs. However, this solution cannot avoid resource waste on nodes. This issue is caused by low resource utilization of some tenants.



Sandboxed-Container allows you to isolate untrusted applications by using sandboxed virtual machines. This prevents the risks of container escapes. This also allows you to deploy different containerized applications on each node. This way, the following benefits are provided:

- Resource scheduling is simplified.
- A node is no longer exclusive to a service. This improves node resource usage and reduces resource fragments and cluster resource costs.
- Sandboxed containers use lightweight virtual machines to provide almost the same performance as containers in runC.



3.1.4.13.8. Differences between runC and runV

This topic describes the differences between runC and Sandboxed-Container (runV) in terms of their performance and pod creation methods. This allows you to better understand and utilize the benefits of sandboxed containers.

Differences between runC and runV

Item	runC	runV
Container engine	Docker and Containerd	Containerd
Node type	Elastic Compute Service (ECS) instances and ECS Bare Metal instances	EBM
Container kernel	Share the host kernel	Dedicated kernel
Container isolation	Cgroups and namespaces	Lightweight virtual machines (VMs)
Rootfs Graph Driver	OverlayFS	DeviceMapper
RootFS I/O throttling	Cgroups	DeviceMapper Block IO Limit **Note** Supported by only Sandboxed-Container V1.
NAS mounting	Not supported	Supported
Disk mounting	Not supported	Supported

Item	runC	runV
Collection of container logs	Logtail directly collects container logs from the host.	logtail sidecar
Pod Overhead	None	• Sandboxed-Container V1: For example, if you set memory: 512 Mi for a pod overhead, it indicates that 512 MiB of memory is allocated to the pod sandbox. Pod overhead refers to the amount of resources consumed by the pod sandbox. In this case, if you set a memory limit of 512 MiB for containers in the pod, the pod will request a total memory of 1,024 MiB. • Sandboxed-Container V2: The memory limit for a pod overhead is calculated based on the following formula: Memory for a pod overhead = 64 MiB + Requested memory of containers in a pod × 2%. If the result is greater than 512 MiB, the value is set to 512 Mi. If the result is smaller than 64 MiB, the value is set to 64 Mi.

Differences in pod creation between runC and runV

You can connect to clusters of Container Service by using kubectl. For more information, see Connect to a cluster through kubectl.

- Create a pod that uses runC
  i. (Optional)Use `runtimeClassName: runc` to set the container runtime to runC.

  > **Note** The preceding command is optional. runC is the default container runtime.

  ii. Run the following commands to create a pod that uses runC:

```
cat <<EOF | kubectl create -f -
apiVersion: v1
kind: Pod
metadata:
  name: busybox-runc
  labels:
    app: busybox-runc
spec:
  containers:
  - name: busybox
    image: registry.cn-hangzhou.aliyuncs.com/acs/busybox:v1.29.2
    command:
    - tail
    - -f
    - /dev/null
    resources:
      limits:
        cpu: 1000m
        memory: 512Mi
      requests:
        cpu: 1000m
        memory: 512Mi
EOF
```

- Create a pod that uses runV
  i. Use `runtimeClassName: runv` to set the container runtime to runV.
  ii. (Optional)Run the following command to verify that a RuntimeClass object named `runv` exists in the cluster.

```
kubectl get runtimeclass runv -o yaml
```

> **Note** A RuntimeClass object named `runv` is automatically created in a Kubernetes cluster that uses Sandboxed-Container.

  iii. Run the following command to create a pod that uses runV:

```
cat <<EOF | kubectl create -f -
apiVersion: v1
kind: Pod
metadata:
  name: busybox-runv
  labels:
    app: busybox-runv
spec:
  runtimeClassName: runv
  nodeSelector:
    alibabacloud.com/container-runtime: Sandboxed-Container.runv
  containers:
  - name: busybox
    image: registry.cn-hangzhou.aliyuncs.com/acs/busybox:v1.29.2
    command:
    - tail
    - -f
    - /dev/null
    resources:
      limits:
        cpu: 1000m
        memory: 512Mi
      requests:
        cpu: 1000m
        memory: 512Mi
EOF
```

Notice If the Kubernetes version is earlier than 1.16, add the following nodeSelector configuration:

```
nodeSelector:
    alibabacloud.com/container-runtime: Sandboxed-Container.runv
```

iv. Run the following command to query the pod that you have created: If the output is `runv`, it indicates that the pod is running in a sandbox.

```
kubectl get pod busybox-runv -o jsonpath={.spec.runtimeClassName}
```

v. Run the following command to log on to the pod and query its CPU and memory specifications:

```
kubectl exec -ti pod busybox-runv /bin/sh
/ # cat /proc/meminfo | head -n1
MemTotal:        1130692 kB
/ # cat /proc/cpuinfo | grep processor
processor    : 0
```

The output shows that the number of CPUs is not the same as that of the host. The total memory is the sum of pod memory and pod overhead. Be aware that the total memory is slightly smaller because the system also consumes some memory.

3.1.4.13.9. Compatibility notes

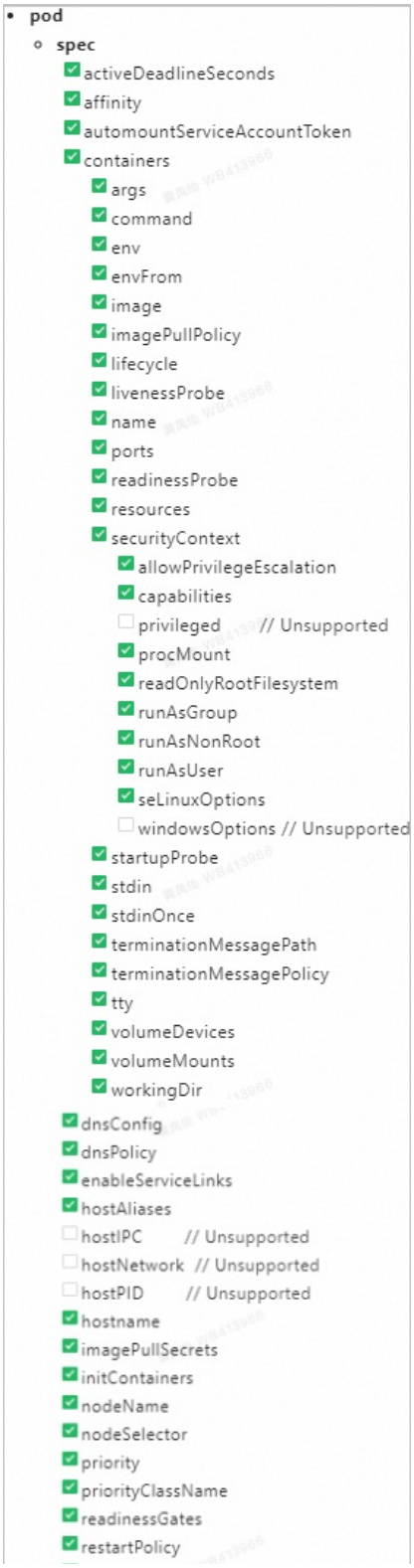
This topic describes the pod fields that are supported by Sandboxed-Container. This allows you to fully use the Sandboxed-Container runtime.

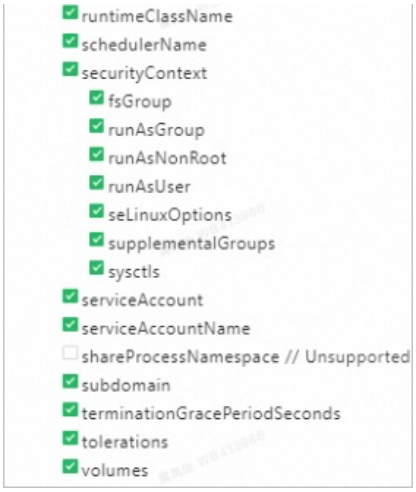
Context

Sandboxed-Container is a new runV container runtime that provides compatibility with runC in terms of pod networking, service networking (ClusterIP and NodePort), and image management. However, Sandboxed-Container does not support all pod fields. To use Sandboxed-Container, you do not need to change your development mode or image packaging method.

Supported pod fields

Sandboxed-Container supports the following pod fields that are marked by ticks:





3.1.4.14. Create a batch release

Context

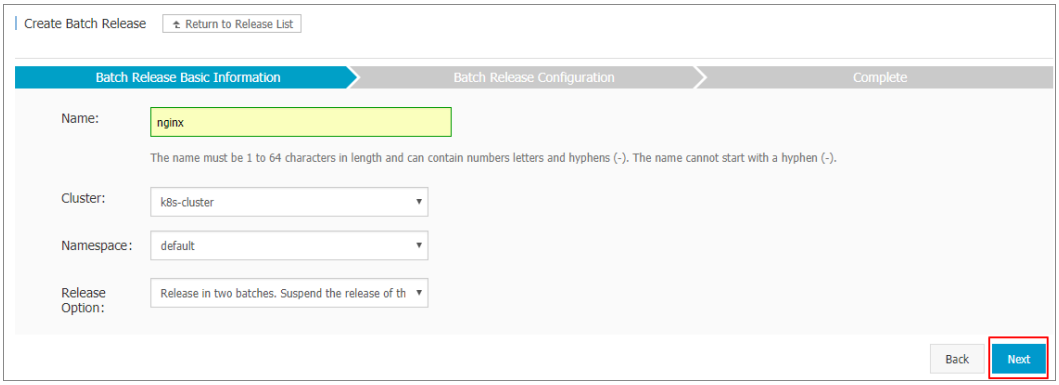
> **Note** In Kubernetes clusters of the latest version, alicloud-application-controller is installed by default. This component is only available in Kubernetes 1.9.3 and later. You can upgrade your cluster through the console.

Procedure

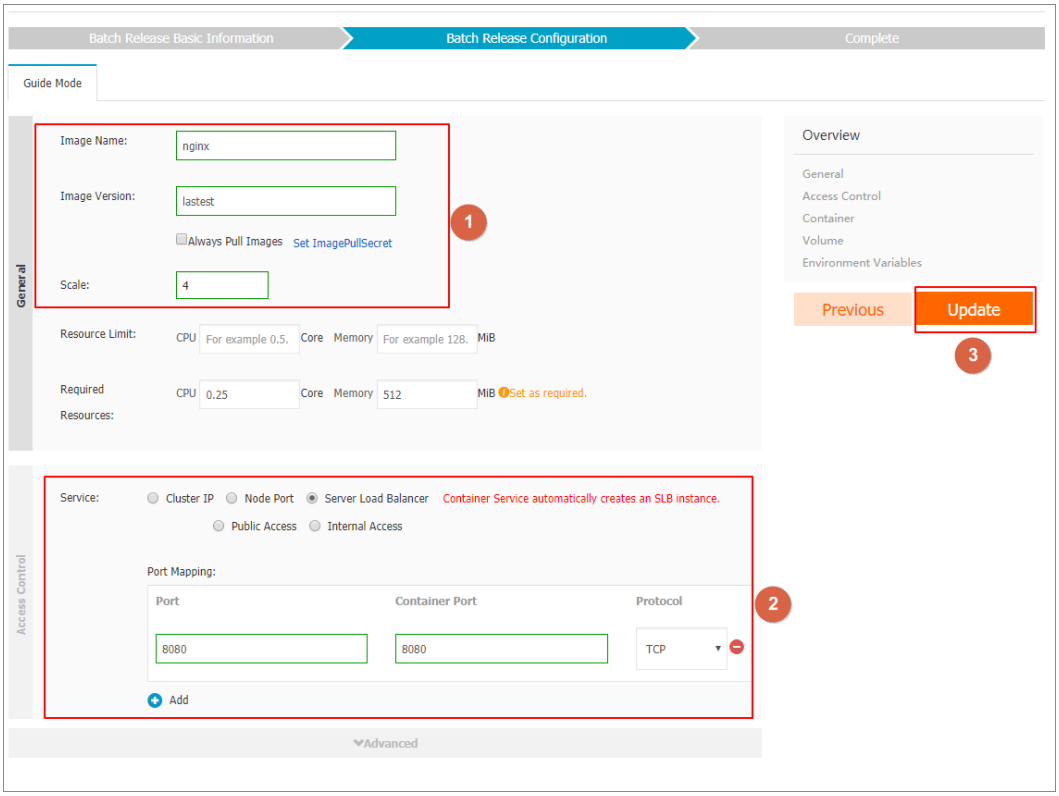
1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Applications > Releases**. Click the **Batch Release** tab and click **Create Batch Release** in the upper-right corner.

   > **Note** If the button is dimmed, it indicates that you need to upgrade your cluster first.

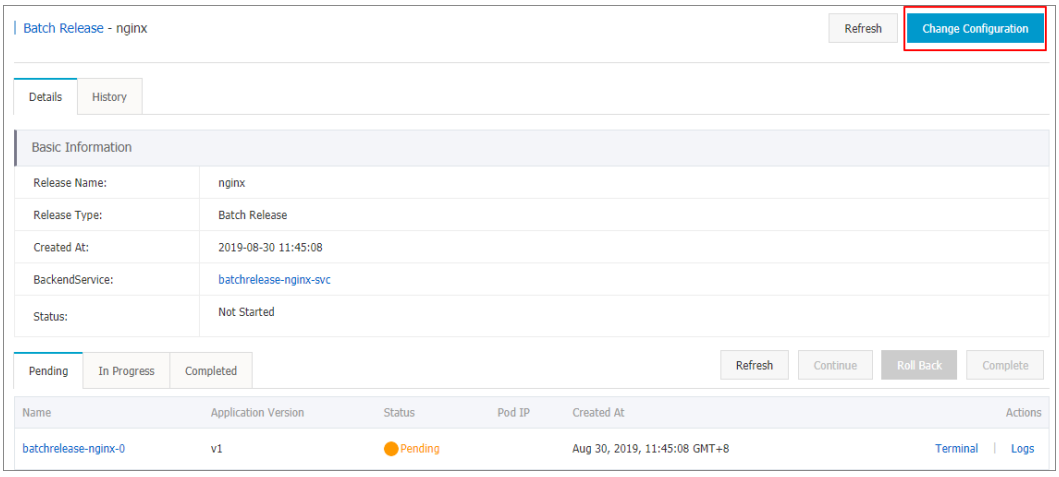
3. On the Batch Release Basic Information tab, set the following parameters: application name, cluster, namespace, and release option. Click **Next**.



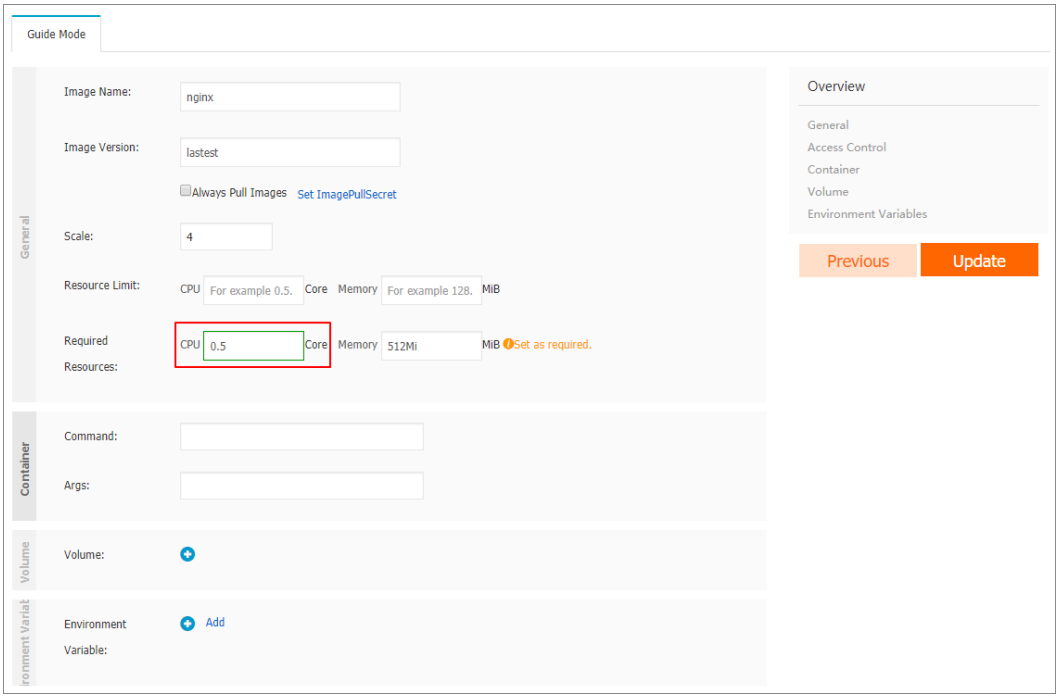
4. On the Batch Release Configuration tab, configure the pods and service, and then click **Update** to create an application.



5. Go to the Releases page and click the Batch Release tab. You can find the newly created application and its status is **Not Started**. Click **Details** in the Actions column.
6. On the Details tab, you can find more information about the application. Click **Change Configuration** in the upper-right corner to change the application configuration.

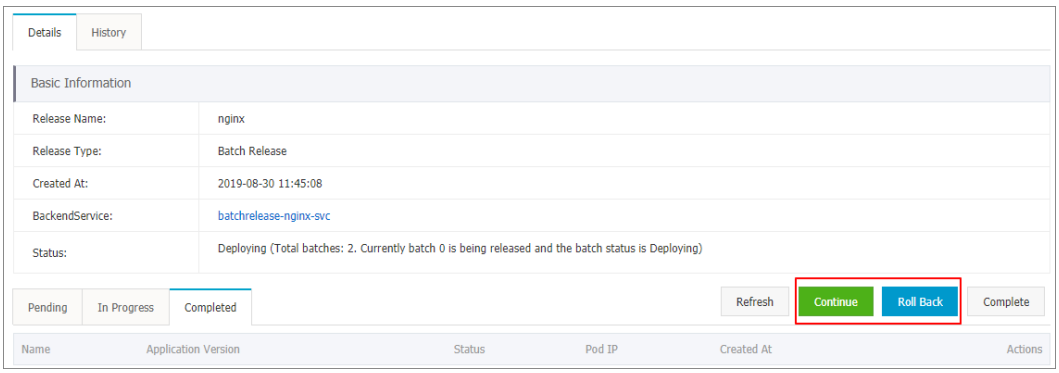


7. On the page that appears, change the configuration and then click **Update**.



8. You are redirected to the Releases page by default. Click the Batch Release tab and you can find the application status. After the first batch is deployed, click **Details**.

9. On the details tab, two pods are listed in the Not Started list and two pods are listed in the Completed list. This indicates that the first batch has been released. Click **Continue** to release the second batch of pods. Click **Roll Back** to roll back to the previous version.



10. After the release is completed, click the **History** tab and you can choose to roll back to a previous version.

What's next

You can create a batch release to quickly verify the functionalities of a new application version while serving all production traffic. A batch release requires less resources than a blue-green release. Currently, you can only create batch releases through the wizard. Support for YAML configuration files will be available soon.

3.1.4.15. Use Log Service to collect container logs

Container Service is integrated with Log Service. When you create a Kubernetes cluster, you can enable Log Service to collect container logs, including standard outputs and text files.

Activate Log Service

To activate Log Service, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console. In the top navigation bar, choose **Products > Log Service** to go to the **Log Service** page.
2. Select the required organization and region.
3. Click **SLS** to go to the Log Service console.

Create a Kubernetes cluster and enable Log Service

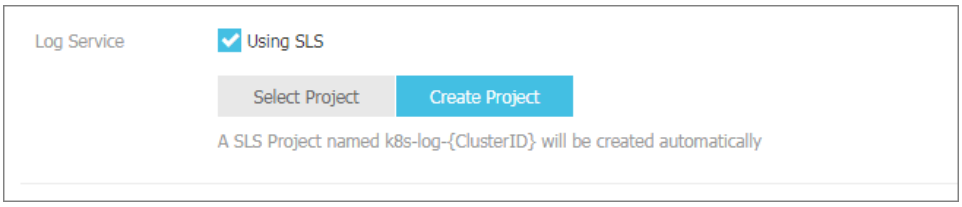
To create a Kubernetes cluster, perform the following steps:

1. Log on to the Container Service console.

   > **Note** The specified organization must be the same as the one you selected when you activate Log Service. For more information, see Activate Log Service.

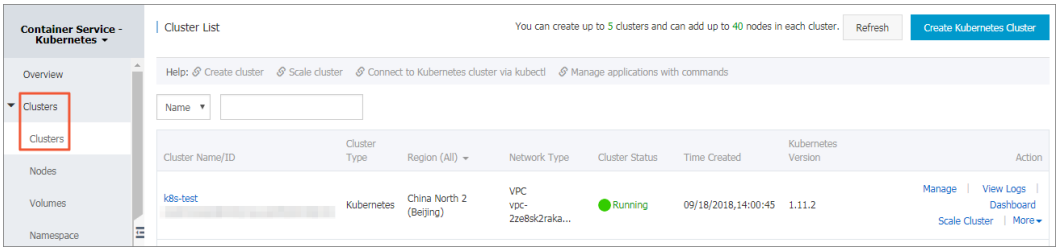
2. In the left-side navigation pane of the Container Service console, choose **Clusters > Clusters**. The Clusters page appears.
3. Click **Create Kubernetes Cluster**. For more information about the configurations, see Create a Kubernetes cluster.
4. In the Component Configuration step, select **Enable Log Service** to install the logging plug-in.
5. After you select the check box, you must specify a project to store log data. You can click **Select Project** and select an existing project from the drop-down list that appears.

   Create a project. By default, the system names the project in the format of `k8s-log-{ClusterID}`. ClusterID indicates the unique ID of the cluster to be created.

6. After you set the other parameters, click **Create Cluster** in the upper-right corner of the page. In the dialog box that appears, click **OK** to create the Kubernetes cluster.

   On the Clusters page, you can find the created cluster.



Install Log Service components in an existing Kubernetes cluster

If you have created a Kubernetes cluster and activated Log Service, you can perform the following steps to use Log Service:

1. Connect to the Kubernetes cluster by using CloudShell.

   For more information, see Connect to a Kubernetes cluster through kubectl.

2. Run the script *logtail-dedicated.sh* to install Log Service components in the Kubernetes cluster.

```
#! /env/bin/bash
yaml=$(cat <<-END
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: alibaba-log-config-file
  namespace: kube-system
data:
  ilogtail_config.json: |
    {
      "config_server_address" : "http://logtail. $REGION.sls-pub. $INTERNET_DOMAIN",
      "data_server_address" : "http://data. $REGION.sls-pub. $INTERNET_DOMAIN",
      "data_server_list" :
      [
          {
              "cluster" : "$REGION",
              "endpoint" : "data. $REGION.sls-pub. $INTERNET_DOMAIN"
          }
      ],
      "shennong_unix_socket" : false
    }
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: alibaba-log-configuration
  namespace: kube-system
data:
    log-project: "k8s-log-$CLUSTER_ID"
    log-endpoint: "data. $REGION.sls-pub. $INTERNET_DOMAIN"
    log-machine-group: "k8s-group-$CLUSTER_ID"
    log-config-path: "/etc/ilogtail/conf/apsara/ilogtail_config.json"
    log-ali-uid: "$ALI_UID"
    log-access-id: "" # just use blank string
    log-access-key: "" # just use blank string
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: alibaba-log-controller
  namespace: kube-system
```

```
  labels:
    k8s-app: alibaba-log-controller
  annotations:
    component.version: "v0.1.3"
    component.revision: "v1"
spec:
  replicas: 1
  template:
    metadata:
      labels:
        k8s-app: alibaba-log-controller
      annotations:
        scheduler.alpha.kubernetes.io/critical-pod: ''
    spec:
      serviceAccountName: alibaba-log-controller
      tolerations:
        - operator: "Exists"
      containers:
      - name: alibaba-log-controller
        image: $IMAGE_REPO_URL/acs/log-controller-$ARCH:v0.1.3.0-527ff4d-aliyun
        resources:
          limits:
            memory: 100Mi
          requests:
            cpu: 50m
            memory: 100Mi
        env:
          - name: "ALICLOUD_LOG_PROJECT"
            valueFrom:
              configMapKeyRef:
                name: alibaba-log-configuration
                key: log-project
          - name: "ALICLOUD_LOG_ENDPOINT"
            valueFrom:
              configMapKeyRef:
                name: alibaba-log-configuration
                key: log-endpoint
          - name: "ALICLOUD_LOG_MACHINE_GROUP"
            valueFrom:
              configMapKeyRef:
                name: alibaba-log-configuration
                key: log-machine-group
          - name: "ALICLOUD_ACS_K8S_FLAG"
            value: "ture"
          - name: "ALICLOUD_ACCESS_KEY_ID"
            valueFrom:
              configMapKeyRef:
                name: alibaba-log-configuration
                key: log-access-id
          - name: "ALICLOUD_ACCESS_KEY_SECRET"
            valueFrom:
              configMapKeyRef:
                name: alibaba-log-configuration
                key: log-access-key
      nodeSelector:
        beta.kubernetes.io/os: linux
---
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
```

```
metadata:
  name: aliyunlogconfigs.log.alibabacloud.com
spec:
  group: log.alibabacloud.com
  version: v1alpha1
  names:
    kind: AliyunLogConfig
    plural: aliyunlogconfigs
  scope: Namespaced
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: alibaba-log-controller
subjects:
- kind: ServiceAccount
  name: alibaba-log-controller
  namespace: kube-system
roleRef:
  kind: ClusterRole
  name: alibaba-log-controller
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
  name: alibaba-log-controller
  labels:
    k8s-app: alibaba-log-controller
rules:
- apiGroups: ["log.alibabacloud.com"]
  resources:
  - aliyunlogconfigs
  verbs:
  - update
  - get
  - watch
  - list
- apiGroups: [""]
  resources:
  - configmaps
  verbs:
  - create
  - update
  - get
- apiGroups: [""]
  resources:
  - events
  verbs:
  - create
  - patch
  - update
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: alibaba-log-controller
  namespace: kube-system
  labels:
    k8s-app: alibaba-log-controller
```

```
    k8s-app: alibaba-log-controller
---
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: logtail-ds
  namespace: kube-system
  labels:
    k8s-app: logtail-ds
  annotations:
    component.version: "v0.16.16"
    component.revision: "v0"
spec:
  updateStrategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        k8s-app: logtail-ds
      annotations:
        scheduler.alpha.kubernetes.io/critical-pod: ''
    spec:
      tolerations:
        - operator: "Exists"
      containers:
      - name: logtail
        image: $IMAGE_REPO_URL/acs/logtail-$ARCH:v0.16.24.0-c46cd2fe-aliyun
        resources:
          limits:
            memory: 512Mi
          requests:
            cpu: 100m
            memory: 256Mi
        livenessProbe:
          exec:
            command:
            - /etc/init.d/ilogtaild
            - status
          initialDelaySeconds: 30
          periodSeconds: 30
        securityContext:
          privileged: false
        env:
          - name: "ALIYUN_LOGTAIL_CONFIG"
            valueFrom:
              configMapKeyRef:
                name: alibaba-log-configuration
                key: log-config-path
          - name: "ALIYUN_LOGTAIL_USER_ID"
            valueFrom:
              configMapKeyRef:
                name: alibaba-log-configuration
                key: log-ali-uid
          - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
            valueFrom:
              configMapKeyRef:
                name: alibaba-log-configuration
                key: log-machine-group
          - name: "ALICLOUD_LOG_DOCKER_ENV_CONFIG"
            value: "true"
```

```
        - name: "ALICLOUD_LOG_ECS_FLAG"
          value: "ture"
        - name: "ALICLOUD_LOG_DEFAULT_PROJECT"
          valueFrom:
            configMapKeyRef:
              name: alibaba-log-configuration
              key: log-project
        - name: "ALICLOUD_LOG_ENDPOINT"
          valueFrom:
            configMapKeyRef:
              name: alibaba-log-configuration
              key: log-endpoint
        - name: "ALICLOUD_LOG_DEFAULT_MACHINE_GROUP"
          valueFrom:
            configMapKeyRef:
              name: alibaba-log-configuration
              key: log-machine-group
        - name: "ALICLOUD_LOG_ACCESS_KEY_ID"
          valueFrom:
            configMapKeyRef:
              name: alibaba-log-configuration
              key: log-access-id
        - name: "ALICLOUD_LOG_ACCESS_KEY_SECRET"
          valueFrom:
            configMapKeyRef:
              name: alibaba-log-configuration
              key: log-access-key
        - name: "ALIYUN_LOG_ENV_TAGS"
          value: "_node_name_|_node_ip_"
        - name: "_node_name_"
          valueFrom:
            fieldRef:
              fieldPath: spec.nodeName
        - name: "_node_ip_"
          valueFrom:
            fieldRef:
              fieldPath: status.hostIP
      volumeMounts:
      - name: sock
        mountPath: /var/run/docker.sock
      - name: root
        mountPath: /logtail_host
        readOnly: true
      - name: alibaba-log-config-file-volume
        mountPath: /etc/ilogtail/conf/apsara
        readOnly: true
    terminationGracePeriodSeconds: 30
    nodeSelector:
      beta.kubernetes.io/os: linux
    volumes:
    - name: sock
      hostPath:
        path: /var/run/docker.sock
        type: Socket
    - name: root
      hostPath:
        path: /
        type: Directory
    - name: alibaba-log-config-file-volume
      configMap:
```

```
        configMap:
          name: alibaba-log-config-file
END
)
echo "$yaml" > logtail.yml
kubectl create -f logtail.yml
```

3. Replace `<your_server_architecture>`, `<your_k8s_cluster_region_id>`, `<your_k8s_cluster_id>`, `<k8s_cluster_domain_suffix>`, `<your_ali_uid>`, and `<your_image_repo_url>` with actual values, and run the following commands. This allows you to set the environment variables and deploy the components.

```
export ARCH=<your_server_architecture>
export REGION=<your_k8s_cluster_region_id>
export CLUSTER_ID=<your_k8s_cluster_id>
export INTERNET_DOMAIN=<k8s_cluster_domain_suffix>
export IMAGE_REPO_URL=<your_image_repo_url>
export ALI_UID=<your_ali_uid>
bash logtail-dedicated.sh // Run the script to install the components
```

> **Note**
>
> - `<your_server_architecture>`: the server architecture, for example, amd64.
> - `<your_k8s_cluster_region_id>`: the region where the Kubernetes cluster is deployed, for example, cn-qingdao-apsara-d01.
> - `<your_k8s_cluster_id>`: the ID of the Kubernetes cluster.
> - `<k8s_cluster_domain_suffix>`: the domain suffix of the Kubernetes cluster, for example, env28.internet.com.
> - `<your_ali_uid>`: the ID of the Apsara Stack tenant account, for example, 1234074238634394.
> - `<your_image_repo_url>`: the URL of the image repository, for example, registry.cn-hangzhou.aliyuncs.com.

Create an application and configure Log Service

When you create an application in Container Service, you can configure Log Service to collect container logs. You can only use YAML templates to configure Log Service.

1. Log on to the Container Service console. In the left-side navigation pane, choose **Applications > Deployments**. In the upper-right corner of the Deployments page, click **Create from Template**.
2. YAML templates follow the Kubernetes syntax. You can use environment variables to add collection configurations and custom tags. You must also configure volumeMounts and volumes. The following sample template describes how to create a pod.

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  labels:
    app: logtail-test
  name: logtail-test
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: logtail-test
      name: logtail-test
    spec:
      containers:
      - name: logtail
        image: registry.acs.env28.intranet.com/acs/busybox:latest
        args:
        - ping
        - 127.0.0.1
        env:
        - name: aliyun_logs_log-stdout
          value: stdout
        - name: aliyun_logs_log-varlog
          value: /log/*.log
        - name: aliyun_logs_log_tags
          value: tag1=v1
        volumeMounts:
        - name: volumn-sls
          mountPath: /log
      volumes:
      - name: volumn-sls
        emptyDir: {}
```

- Specify the following configurations in order based on your business requirements:
- Use environment variables to add collection configurations and custom tags. All environment variables for collection configurations must be prefixed with `aliyun_logs_` .
- Add collection configurations in the following format:

  ```
  - name: aliyun_logs_{Logstore name}
    value: {Log path}
  ```

  The preceding YAML template uses two environment variables to add collection configurations. The environment variable `aliyun_logs_log-stdout` indicates that a Logstore named log-stdout is created and standard outputs of container logs are stored in the log-stdout Logstore.

  > **Note** The name of a Logstore can contain hyphens (-) but cannot contain underscores (_).

- **Custom tags** are configured in the following format:

  ```
  - name: aliyun_logs_{Tag name without underscores (_)}_tags
    value: {Tag name}={Tag value}
  ```

  After a tag is added, the tag is automatically appended to log fields when logtail-ds collects the container logs.
- If you specify a log path to collect logs that are not standard outputs, you must configure volumeMounts.

In the preceding YAML template, the mountPath field in volumeMounts is set to */var/log*. This allows logtail-ds to collect */var/log/*.log* files.

3. After you edit the YAML template, click **Create**.

Set advanced parameters

You can set more environment variables to apply advanced configurations to log collection. The following table provides details of these variables.

Variable	Description	Example	Note

aliyun_logs_{key}	• Required. The key field can contain lowercase letters, digits, and hyphens (-), and cannot contain underscores (_).
	• If the environment variable aliyun_logs_{key}_logstore is not set, a Logstore named {key} is created to collect logs.
	• To collect standard outputs of container logs, set the value to stdout. You can also set the value to a path inside a container.
		`- name: aliyun_logs_catalina`
		`stdout`
		
		`- name: aliyun_logs_access-log`
		
		`/var/log/nginx/access.log`
			• By default, the simple mode is used to collect logs. To parse log data, we recommend that you use the Log Service console.
			• The value of {key} must be unique in the cluster.

aliyun_logs_{key}_tags	Optional. This variable is used to add marks to log data. It must be in the {tag-key}={tag-value} format.	`- name: aliyun_logs_catalina_tags` `app=catalina`	-
aliyun_logs_{key}_project	Optional. This variable specifies a project in Log Service. By default, the project that you specified when you create the Kubernetes cluster is used.	`- name: aliyun_logs_catalina_project` `my-k8s-project`	The region of the project must be the same as where your logtail-ds is located.
aliyun_logs_{key}_logstore	Optional. This variable specifies a Logstore in Log Service. By default, the Logstore is named after {key}.	`- name: aliyun_logs_catalina_tags` `my-logstore`	-

Variable	Description	Example	Note
aliyun_logs_{key}_shard	Optional. This variable specifies the number of shards in the Logstore. Valid values: 1 to 10. Default value: 2.	- name: aliyun_logs_catalina_shard 4	-
aliyun_logs_{key}_ttl	Optional. This variable specifies the number of days for which log data is retained. Valid values: 1 to 3650. • To permanently retain log data, set the value to 3650. • Default value: 90.	- name: aliyun_logs_catalina_ttl 3650	-
aliyun_logs_{key}_machinegroup	Optional. This permanently specifies the machine group of the application. By default, the machine group is the one where your logtail-ds is located.	- name: aliyun_logs_catalina_machinegroup my-machine-group	-

- Scenario 1: Collect logs from multiple applications and store the logs in the same Logstore

  In this scenario, set the aliyun_logs_{key}_logstore parameter. The following example shows how to collect standard outputs from two applications and store the logs in stdout-logstore.

  Set the following environment variables for Application 1:

```
######### Set environment variables ###########
    - name: aliyun_logs_app1-stdout
      value: stdout
    - name: aliyun_logs_app1-stdout_logstore
      value: stdout-logstore
```

  Set the following environment variables for Application 2:

```
######### Set environment variables ###########
    - name: aliyun_logs_app2-stdout
      value: stdout
    - name: aliyun_logs_app2-stdout_logstore
      value: stdout-logstore
```

- Scenario 2: Collect logs from different applications and store the logs in different projects

  In this scenario, perform the following steps:

  i. Create a machine group in each project and set the machine group ID in the following format: k8s-group-{cluster-id}, where {cluster-id} is the ID of the Kubernetes cluster. You can customize machine group names.

  ii. Specify the project, Logstore, and machine group in the environment variables for each application.

```
######### Set environment variables ###########
    - name: aliyun_logs_app1-stdout
      value: stdout
    - name: aliyun_logs_app1-stdout_project
      value: app1-project
    - name: aliyun_logs_app1-stdout_logstore
      value: app1-logstore
    - name: aliyun_logs_app1-stdout_machinegroup
      value: app1-machine-group
```

View logs

To view log data, perform the following steps:

1. Log on to the Log Service console. For more information, see Activate Log Service.
2. Select the required project. The default project ID is k8s-log-{Cluster ID}.
3. In the list of Logstores, find the Logstores and click **Search** in the Log Search column for each Logstore. In this example, the Logstores are log-stdout and log-varlog.
4. On the Raw Logs tab, you can view raw logs in log-stdout and log-varlog.

4.Auto Scaling (ESS)

4.1. User Guide

4.1.1. What is Auto Scaling?

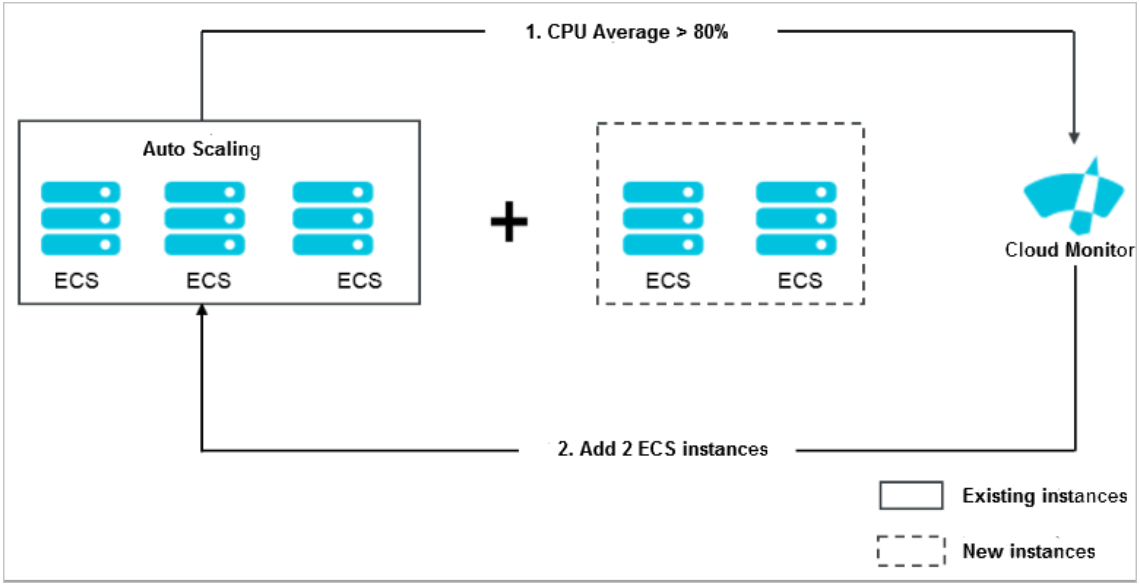
Auto Scaling is a management service that automatically adjusts your elastic computing resources based on your business needs and policies.

When business loads increase, Auto Scaling automatically adds ECS instances based on user-defined scaling rules to ensure sufficient computing capabilities. When business loads decrease, Auto Scaling automatically removes ECS instances to save costs.

Auto Scaling provides the following features:

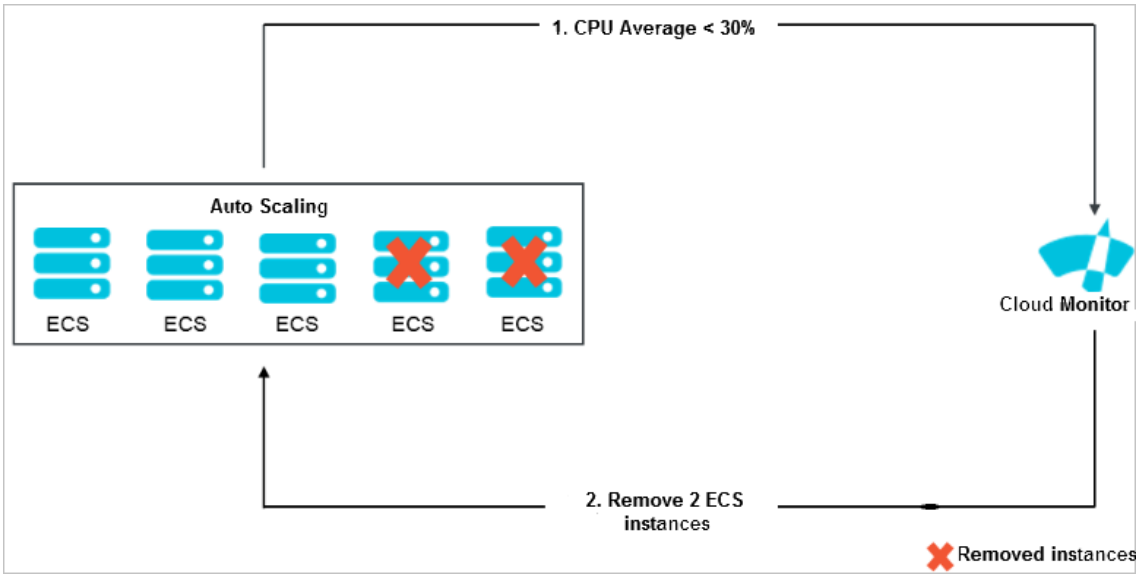
- Scale-out

  When business loads surge above normal loads, Auto Scaling automatically increases underlying resources. This helps maintain access speed and ensures that resources are not overloaded. For example, if the CPU utilization of ECS instances exceeds 80%, Auto Scaling scales out ECS resources based on your configured rules. During the scale-out event, Auto Scaling automatically creates and adds ECS instances to a scaling group, and adds the new instances to the backend server groups of the associated SLB instances and the whitelists of the associated ApsaraDB RDS instances. The following figure shows the implementation of a scale-out event.



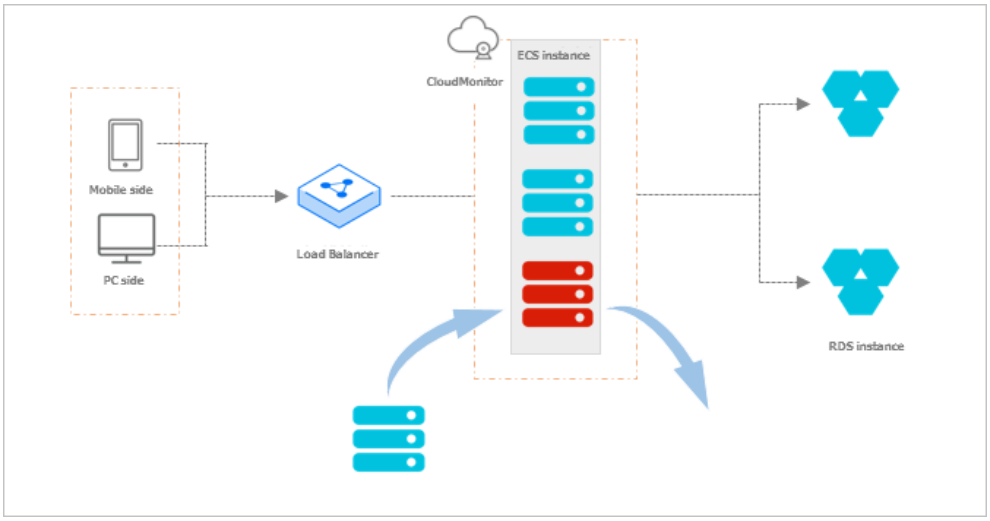
- Scale-in

  When your business loads decrease, Auto Scaling automatically releases underlying resources to prevent resource wastage and reduce costs. For example, if the CPU utilization of ECS instances in a scaling group is less than 30%, Auto Scaling automatically scales in ECS instances based on your configured rules. During the scale-in event, Auto Scaling removes ECS instances from the scaling group and also from the backend server groups of the associated SLB instances as well as the whitelists of the associated ApsaraDB RDS instances. The following figure shows the implementation of a scale-in event.



- Elastic recovery

  If ECS instances in a scaling group are not in the Running state, Auto Scaling considers the instances to be unhealthy. If an ECS instance is considered unhealthy, Auto Scaling automatically releases the instance and creates a new one. This process is called elastic recovery. It ensures that the number of healthy ECS instances in a scaling group does not fall below the minimum number of ECS instances that you specified for the scaling group. The following figure shows the implementation of elastic recovery.



4.1.2. Notes

4.1.2.1. Precautions

This topic describes the precautions when you use Auto Scaling (ESS).

Scaling rules

ESS uses scaling rules to scale ECS instances in a scaling group based on the minimum and maximum numbers of ECS instances specified for the scaling group. Assume that a scaling group can contain up to 45 ECS instances. If you configure a scaling rule to increase the number of ECS instances in the scaling group to 50, ESS only increases the number of ECS instances to 45 at most.

Scaling activities

- Only one scaling activity can be executed at a time in a scaling group.
- An ongoing scaling activity cannot be terminated. For example, if a scaling activity is being executed to create 20 ECS instances but only five have been created, you cannot forcibly terminate the scaling activity.
- If some ECS instances fail to be added to a scaling group during a scaling activity, ESS considers that the scaling activity is complete without trying to add the failed instances to the scaling group. ESS rolls back the ECS instances that fails to be added but not the scaling activity. For example, if ESS has created 20 ECS instances for a scaling group, and 19 of the instances are added to SLB instances, only the one ECS instance that failed to be added is automatically released.

Cooldown period

- During the cooldown period, if you manually execute a scaling task, such as a scaling rule or scheduled task, the task is immediately executed without waiting for the cooldown period to expire.
- The cooldown period starts after the last ECS instance is added to or removed from a scaling group during a scaling activity.

4.1.2.2. Manual intervention

If you manually intervene with Auto Scaling operations, Auto Scaling processes the intervention accordingly.

Auto Scaling does not prevent you from performing manual intervention, such as deleting automatically created ECS instances in the ECS console. The following table describes how Auto Scaling processes manual intervention.

Resource	Manual intervention type	Processing method
ECS	A user deletes an ECS instance from a scaling group by using the ECS console or calling API operations.	Auto Scaling performs health checks to determine whether the ECS instance is unhealthy. If the instance is unhealthy, Auto Scaling removes it from the scaling group. The internal IP address of the ECS instance is not automatically deleted from the whitelist of the associated ApsaraDB RDS (RDS) instance. If the total number of ECS instances in the scaling group falls below the lower limit after the ECS instance is removed, the scaling group automatically creates an ECS instance to ensure that the number of instances reaches the lower limit.
ECS	A user revokes the ECS API permissions granted to Auto Scaling.	Auto Scaling rejects all scaling activity requests.
SLB	A user manually removes an ECS instance from an SLB instance by using the SLB console or calling API operations.	Auto Scaling does not automatically detect this action or handle such exceptions. The ECS instance remains in the scaling group. When a scale-in event is triggered, Auto Scaling releases the ECS instance if the instance meets the removal policy.
SLB	A user manually deletes an SLB instance or disables the health check feature for an SLB instance by using the SLB console or calling API operations.	Auto Scaling does not add ECS instances to scaling groups that are associated with this SLB instance. Auto Scaling removes ECS instances from the scaling groups if a scaling task triggers a scale-in rule or the ECS instances are recognized as unhealthy after a health check is performed.
SLB	An SLB instance is unavailable because of system-related reasons.	All scaling activities fail except for instance removal tasks that are manually executed.

Resource	Manual intervention type	Processing method
SLB	A user revokes the SLB API permissions granted to Auto Scaling.	Auto Scaling rejects all scaling activity requests for scaling groups with which SLB instances are associated.
RDS	A user manually removes the IP address of an ECS instance from the whitelist of the associated RDS instance by using the RDS console or calling API operations.	Auto Scaling does not automatically detect this action or handle such exceptions. The ECS instance remains in the scaling group. When a scale-in event is triggered, Auto Scaling releases the ECS instance if the instance meets the removal policy.
RDS	A user manually deletes an RDS instance by using the RDS console or calling API operations.	Auto Scaling does not add ECS instances that are associated with this RDS instance to scaling groups. Auto Scaling removes ECS instances from the scaling groups if a scaling task triggers a scale-in rule or the ECS instances are recognized as unhealthy after a health check is performed.
RDS	An RDS instance is unavailable because of system-related reasons.	All scaling activities fail except for instance removal tasks that are manually executed.
RDS	A user revokes the RDS API permissions granted to Auto Scaling.	Auto Scaling rejects all scaling activity requests for the scaling groups associated with RDS instances.

4.1.2.3. Limits

This topic describes the limits of ESS.

- ESS does not support vertical scaling. It can only scale the number of ECS instances. The CPU, memory, and bandwidth configurations of ECS instances cannot be automatically adjusted.
- The following table describes the quantity limits that are applied to a scaling group.

Item	Quota
Scaling configuration	You can create a maximum of 10 scaling configurations for a scaling group.
Scaling rule	You can create a maximum of 50 scaling rules for a scaling group.
ECS instance	A scaling group can contain a maximum of 1,000 ECS instances.

4.1.2.4. Scaling group status

This topic describes the states of a scaling group in the console and in an API operation.

State in the console	State in an API operation
Creating	Inactive
Created	Inactive
Enabling	Inactive
Enabled	Active
Disabling	Inactive

State in the console	State in an API operation
Disabled	Inactive
Deleting	Deleting

4.1.2.5. Scaling processes

Before you use Auto Scaling, you must understand the processes related to scaling activities.

Automatic scaling of a scaling group

- Automatic scale-out
  i. Check the health status and boundary conditions of the scaling group.
  ii. Assign the activity ID and execute the scaling activity.
  iii. Create ECS instances.
  iv. Modify Total Capacity.
  v. Assign IP addresses to the created ECS instances.
  vi. Add the ECS instances to the whitelist of the associated ApsaraDB RDS instance.
  vii. Start the ECS instances.
  viii. Associate the ECS instances with an SLB instance and set the weight to the SLB weight value that is specified when the scaling configuration is created.
  ix. The cooldown period starts after the scaling activity is complete.
- Automatic scale-in
  i. Check the health status and boundary conditions of the scaling group.
  ii. Assign the activity ID and execute the scaling activity.
  iii. Remove ECS instances from the associated SLB instance.
  iv. Stop the ECS instances.
  v. Remove the ECS instances from the whitelist of the associated ApsaraDB RDS instance.
  vi. Release the ECS instances.
  vii. Modify Total Capacity.
  viii. The cooldown period starts after the scaling activity is complete.

Manually add or remove existing ECS instances

- Manually add instances
  i. Check the health status and boundary conditions of the scaling group, and check the status and type of ECS instances.
  ii. Assign the activity ID and execute the scaling activity.
  iii. Add the ECS instances.
  iv. Modify Total Capacity.
  v. Add the ECS instances to the whitelist of the associated ApsaraDB RDS instance.
  vi. Associate the ECS instances with an SLB instance and set the weight to the SLB weight value that is specified in the active scaling configuration.

> **Note** If you want to manually add an instance to a scaling group, the instance type of the instance must be the same as that specified in the active scaling configuration of the scaling group. Therefore, you must set the weight to the SLB weight value that is specified in the active scaling configuration.

vii. The cooldown period starts after the scaling activity is complete.

- Manually remove instances
  i. Check the health status and boundary conditions of the scaling group.
  ii. Assign the activity ID and execute the scaling activity.
  iii. SLB stops forwarding traffic to ECS instances.
  iv. Remove the ECS instances from SLB after 60 seconds.
  v. Remove the ECS instances from the whitelist of the associated ApsaraDB RDS instance.
  vi. Modify Total Capacity.
  vii. Remove the ECS instances from the scaling group.
  viii. After the scaling activity is complete, the cooldown period starts.

4.1.2.6. Remove unhealthy ECS instances

Before you use ESS, you must understand information about the removal of unhealthy ECS instances.

After an ECS instance is added to a scaling group, ESS checks the status of the instance on a regular basis. If the ECS instance is not in the Running state, ESS removes the ECS instance from the scaling group. The removal method depends on how the ECS instance is added:

- If an ECS instance is automatically created, ESS immediately removes and releases it.
- If an ECS instance is manually added, ESS immediately removes it, but does not stop or release it.

The removal of unhealthy ECS instances is not limited by the MinSize value. After the unhealthy ECS instances are removed, the number of ECS instances (Total Capacity) may fall below the MinSize value. In this case, ESS automatically creates ECS instances based on the difference between the actual instance number and MinSize value to ensure that the total number of ECS instances is equal to the MinSize value.

4.1.2.7. Instance rollback after a failed scaling activity

Before you use ESS, you must understand the mechanism of instance rollback after a failed scaling activity.

If some ECS instances fail to be added to a scaling group during a scaling activity, ESS considers that the scaling activity is complete without trying to add the failed instances to the scaling group. ESS rolls back ECS instances, not the scaling activity.

For example, if a scaling group has created 20 ECS instances, and 19 of the instances are added to SLB instances, only the one ECS instance that failed to be added is automatically released.

4.1.2.8. Instance lifecycle management

Before you use Auto Scaling (ESS), you must understand concepts related to the instance lifecycle.

Automatically created ECS instances

ECS instances are automatically created by ESS based on user-defined scaling configurations and rules.

ESS manages the entire lifecycle of automatically created ECS instances. ESS creates ECS instances during scale-out events, and stops and releases them during scale-in events.

Manually added ECS instances

ECS instances are manually added to a scaling group.

ESS does not manage the entire lifecycle of manually added ECS instances. These instances are not automatically created by ESS, but are manually added by a user to a scaling group. If the ECS instances are manually or automatically removed from the scaling group, ESS removes the instances but does not stop or release them.

Instance status

An ECS instance in a scaling group goes through the following states during its lifecycle:

- Pending: The ECS instance is being added to the scaling group. The instance is being created, added to an SLB instance, or added to the whitelist of the associated ApsaraDB RDS instance.
- InService: The ECS instance is added to the scaling group and is providing services normally.
- Removing: The ECS instance is being removed from the scaling group.

Instance health status

An ECS instance in a scaling group has the following health states:

- Healthy
- Unhealthy

If an ECS instance is not in the Running state, ESS considers the instance to be unhealthy and automatically removes it from the scaling group.

- ESS stops and releases automatically created ECS instances.
- ESS does not stop or release manually added ECS instances.

4.1.3. Quick start

4.1.3.1. Overview

This topic describes how to create a scaling group and how to add or remove ECS instances.

You can perform the following steps to create a scaling group, and add or remove ECS instances.

1. Create a scaling group

   Set the parameters for the scaling group, such as the Maximum Capacity and Minimum Capacity of ECS instances.

2. Create a scaling configuration

   Set the parameters for the scaling configuration, such as Instance Type and Image.

3. Enable a scaling group

   Enable the scaling group after creating the scaling configuration.

4. Create a scaling rule

   Specify how to add or remove ECS instances. For example, add an ECS instance to a scaling group.

5. Create a scheduled task

   Create scheduled tasks to add or remove instances at a specified time point. Auto Scaling executes the scheduled tasks and scaling rules at the specified time. For example, Auto Scaling can trigger a task to execute a specific scaling rule at 08:00 everyday.

4.1.3.2. Log on to the Auto Scaling console

This topic describes how to log on to the Auto Scaling console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Login**.
4. In the top navigation bar, choose **Products > Elastic Computing > Auto Scaling**.

4.1.3.3. Create a scaling group

This topic describes how to create a scaling group. A scaling group is a group of ECS instances that is dynamically scaled based on the configured scenario. You can specify the minimum and maximum numbers of ECS instances in a scaling group.

Prerequisites

- A VPC and a vSwitch are created. For more information, see Create a VPC and Create a vSwitch in *VPC User Guide*.
- To associate a scaling group with SLB instances, make sure that the following requirements are met:
  - You have one or more SLB instances in the **Running** state.
  - The SLB instances and the scaling group are in the same organization, resource set, and region.
- To associate a scaling group with ApsaraDB RDS (RDS) instances, make sure that the following requirements are met:
  - You have one or more RDS instances in the **Running** state.
  - The RDS instances and the scaling group are in the same organization, resource set, and region.

Procedure

1.
2.
3.
4. In the upper-right corner of the Scaling Groups page, click **Create Scaling Group**.
5. Configure parameters for the scaling group.

Parameter	Required	Description

Parameter	Required	Description
Scaling Group	Yes	The name of the scaling group. It must be 2 to 64 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or a digit.
Organization/Resource Group	Yes	The organization and resource set to which the scaling group belongs.
Maximum Capacity	Yes	The maximum number of instances that a scaling group can contain. Set the value based on business needs to control costs. Valid values: 0 to 1000.
Minimum Capacity	Yes	The minimum number of instances that a scaling group must contain. Set the value based on business needs to ensure service availability. When the scaling group is enabled, Auto Scaling automatically creates this number of ECS instances. Valid values: 0 to 1000.
Cooldown (Seconds)	Yes	The time during which Auto Scaling cannot execute any new scaling activities. The cooldown time occurs after the scaling group successfully executes a scaling activity. During the cooldown time, Auto Scaling rejects all scaling activity requests triggered by event-triggered tasks from Cloud Monitor. However, scaling activities triggered by other types of tasks such as manually triggered tasks and scheduled tasks are not limited by the cooldown time. These tasks are immediately executed. The value must be an integer that is greater than or equal to zero. Unit: seconds.

Parameter	Required	Description
Scale-In Policy	No	The policy for automatically removing ECS instances from the scaling group. This parameter contains the **Select** and **From the list, select** fields. You cannot specify the same values for the two fields. Valid values: ○ **Earliest Instance Created Using Scaling Configuration**: filters instances that were created based on the earliest scaling configuration. ○ **Earliest Created Instance**: filters instances that were added to the scaling group at the earliest point in time. ○ **Newest Instances**: filters instances that were most recently added to the scaling group. ○ **None**: is available only for **From the list, select**. This value indicates that Auto Scaling does not filter instances based on the From the list, select field. For example, if Auto Scaling filters instances based on the **Earliest Created Instance** value of Select, you can select one of the following values for From the list, select: ○ **None**: indicates that Auto Scaling does not filter instances based on the From the list, select field. ○ **Newest Instances**: filters instances obtained based on the Select field and then filters instances that were most recently added to the scaling group.
Region/VPC	Yes	The region and VPC to which the scaling group belongs.
VSwitch	Yes	The ID of the vSwitch with which the scaling group is associated.
Associate SLB Instance	No	After you associate SLB instances with the scaling group, ECS instances that are added to the scaling group are automatically added as SLB backend servers. You can specify a server group to which to add the ECS instances. ECS instances can be added to the following types of server groups: ○ Default server group: the group of ECS instances that are used to receive requests. If the listener is not configured with a vServer group or a primary/secondary server group, requests are forwarded to the ECS instances in the default server group. ○ vServer group: If you want to distribute different requests to different backend servers or configure domain name- or URL-based routing methods, you can use vServer groups.
Associate RDS Instance	No	After you associate RDS instances with the scaling group, the internal IP addresses of ECS instances that are added to the scaling group are automatically added to the whitelists of the RDS instances to allow internal communication.

6. Click **OK**.

Result

The created scaling group is displayed in the scaling group list but is in the **Disabled** state. To enable the scaling group, you must create a scaling configuration. For more information, see Create a scaling configuration.

4.1.3.4. Create a scaling configuration

This topic describes how to create a scaling configuration for a scaling group.

Prerequisites

At least one security group is available. If you do not have any security groups, create a security group. For more information, see Create a security group *in ECS User Guide.*

Context

You can create only a limited number of scaling configurations for a scaling group. For more information, see the Limits topic in *Auto Scaling Product Introduction.*

Procedure

1.
2.
3.
4.
5. Choose **Create > Create Scaling Configuration**.
6. Configure parameters for the scaling configuration.

Section	Parameter	Required	Description
Region	Region	Yes	The region where the ECS instance is located.
	Zone	Yes	The zone where the ECS instance is located.
Security Group	Security Group	Yes	The security group to which the ECS instance belongs.
Instance	Instance Family	Yes	The instance family to which the ECS instance belongs.
	Instance Type	Yes	The instance type of the ECS instance.
Image	Image Type	Yes	◦ **Public Image**: Public images provided by Alibaba Cloud are fully licensed to offer a secure and stable operating environment for applications on ECS instances. ◦ **Custom Image**: You can create custom images to install software or deploy projects that have special requirements.

Section	Parameter	Required	Description
Storage	System Disk	Yes	Specify the category and size of the system disk. The operating system is installed on the system disk. You can select **Ultra Disk** or **Standard SSD**.
	Data Disk	No	Specify the category and size of the data disk. You can select **Ultra Disk** or **Standard SSD**. You can add a maximum of 16 data disks. The maximum capacity of each data disk is 32 TiB. You can set **Release with Instance** and **Encrypt** for each data disk.
Password	Set Password	Yes	Select when to set password. You can select **Now** or **Later**. If you select Later, you can use the Change Password feature in the console to set the password. For more information, see the Change Password topic in *ECS User Guide*.
	Logon Password	No	The password used to log on to the ECS instance. The password must be 8 to 30 characters in length and must contain at least three of the following character types: digits, uppercase letters, lowercase letters, and special characters. **Note** The password is used to log on to the operating system and is not the VNC password.
	Confirm Password	No	Enter the password again.
Deployment Set	Deployment Set	No	The deployment set to which the instance belongs.
Instance Name	Configuration Name	No	The name of the scaling configuration.
	Instance Name	No	The name of the ECS instance.
User Data	User Data	No	Windows supports two formats: Bat and Powershell. Before you perform Base64 encoding, make sure to include `[bat]` or `[powershell]` as the first line. You can run shell scripts for Linux ECS instances.

Section	Parameter	Required	Description
Quantity	Quantity	No	The number of instances to purchase.

7. Click **Submit**.

Result

After the scaling configuration is created, it is in the **Disabled** state and is displayed in your scaling configuration list. To automatically create ECS instances, you must apply a scaling configuration. For more information, see Apply a scaling configuration.

4.1.3.5. Enable a scaling group

This topic describes how to enable a scaling group. You can enable a scaling group to trigger scaling activities.

Prerequisites

- The scaling group is in the **Disabled** state.
- The scaling group has a scaling configuration that is in the **Enable** state.

Procedure

1.
2.
3. Find the target scaling group and click **Enable** in the **Actions** column.
4. Click **OK**.

Result

In the **Status** column, the state of the scaling group is changed from **Disabled** to **Enable**.

4.1.3.6. Create a scaling rule

This topic describes how to create a scaling rule. You can create scaling rules to add or remove ECS instances. For example, you can add an ECS instance to a scaling group.

Context

- You can create only a limited number of scaling rules for a scaling group. For more information, see the Limits topic in *Auto Scaling Product Introduction.*
- After a scaling rule is executed, the resulting number of ECS instances in the scaling group may fall outside of the specified range. In this case, Auto Scaling automatically adjusts the number of ECS instances to ensure that the number of ECS instances in the scaling group is within the specified range.

Procedure

1.
2.
3.
4.
5. Click **Create Scaling Rule**.
6. Configure parameters for the scaling rule.

Parameter	Required	Description
Rule Name	Yes	The name of the scaling rule. It must be 2 to 64 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.
Scaling Activity	Yes	The operation that is performed when the scaling rule is triggered. The operations include: ○ **Change to N instances**: After the scaling rule is executed, the number of instances in the scaling group is changed to N. ○ **Add N instances**: After the scaling rule is executed, N instances are added to the scaling group. ○ **Remove N instances**: After the scaling rule is executed, N instances are removed from the scaling group.
Default Cooldown (Seconds)	No	The cooldown period. If this parameter is not specified, the default value is used.

7. Click **OK**.

4.1.3.7. Create a scheduled task

This topic describes how to create a scheduled task to scale computing resources in response to predictable business changes in the future. Scheduled tasks enable the system to automatically obtain sufficient computing resources before business peaks and release idle computing resources after the business peaks.

Context

A scheduled task is preconfigured to execute the specified scaling rule at the specified time. When the specified time arrives, the scheduled task automatically scales computing resources. This allows you to reduce costs and meet business requirements. You can also specify the recurrence for scheduled tasks if business changes are regular.

If multiple scheduled tasks need to be executed in 1 minute, Auto Scaling executes the most recently created scheduled task.

Procedure

1.
2.
3.
4. In the upper-right corner of the Scheduled Tasks page, click **Create Scheduled Task**.
5. In the dialog box that appears, configure parameters for the scheduled task.

Parameter	Required	Description
Task Name	Yes	The name of the scheduled task. The name must be 2 to 64 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.
Description	Yes	The description of the scheduled task.

Parameter	Required	Description
Organization/Resource Group	Yes	The organization and resource set to which the scheduled task belongs.
Start Time	Yes	The time to execute the scheduled task.
Scaling Rules	Yes	The scaling group to be monitored and the scaling rule to be executed.
Retry Interval (Seconds)	No	The period of time during which the system retires to execute the scheduled task. Unit: seconds. If a scaling activity fails to be executed at the specified time, Auto Scaling executes the scheduled task again within the period of time that is specified by the Retry Interval (Seconds) parameter.
Recurrence Settings (Advanced)	No	This parameter specifies whether to repeatedly execute the scheduled task. Select **Recurrence Settings (Advanced)** and set the Recurrence and Expire parameters. The recurrence values include **Daily**, **Weekly**, and **Monthly**.

6. Click **OK**.

Result

The scheduled task that you created is displayed in the scheduled task list.

4.1.3.8. Create an event-triggered task

This topic describes how to create an event-triggered task associated with monitoring metrics in response to emergent or unpredictable business changes. After you create and enable an event-triggered task, Auto Scaling collects data for the specified metric in real time and triggers an alert when the specified condition is met. Then, Auto Scaling executes the scaling rule to scale ECS instances in the scaling group.

Procedure

1.
2.
3.
4. In the upper-right corner of the Event-Triggered Tasks page, click **Create Alert**.
5. In the dialog box that appears, configure parameters for the event-triggered task.

Parameter	Required	Description
Task Name	Yes	The name of the event-triggered task. It must be 2 to 64 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or a digit.
Description	No	The description of the event-triggered task.
Organization/Resource Group	Yes	The organization and resource set in which to create the event-triggered task.
Monitoring Metrics/Scaling Rules	Yes	The scaling group to be monitored and the scaling rule to be executed.
Monitoring Type	Yes	**System-Level Monitoring** is selected by default.

Parameter	Required	Description
Monitoring Metrics	Yes	The metrics that you want to monitor. Valid values: ○ **Average CPU Utilization** ○ **Memory Usage** ○ **Outbound Traffic** ○ **Inbound Traffic**
Monitoring Period	Yes	The period during which data is aggregated and analyzed. The shorter the period, the higher the frequency that the alert is triggered. Unit: minutes. Valid values: ○ **1** ○ **2** ○ **5** ○ **15**
Statistic	Yes	The rule that determines whether to trigger an alert. Select **Average**, **Max Capacity**, or **Min Capacity**, and specify a threshold value. For example, to trigger an alert when the CPU utilization exceeds 80%: ○ Average: An alert is triggered when the average CPU utilization of all ECS instances in the scaling group exceeds 80%. ○ Max Capacity: An alert is triggered when the highest CPU utilization among the ECS instances in the scaling group exceeds 80%. ○ Min Capacity: An alert is triggered when the lowest CPU utilization among the ECS instances in the scaling group exceeds 80%.
Trigger After	Yes	The number of consecutive times that the threshold must be exceeded before the alert is triggered. Valid values: ○ **1** ○ **2** ○ **3** ○ **5**

6. Click **OK**.

4.1.4. Scaling groups

4.1.4.1. Create a scaling group

This topic describes how to create a scaling group. A scaling group is a group of ECS instances that is dynamically scaled based on the configured scenario. You can specify the minimum and maximum numbers of ECS instances in a scaling group.

Prerequisites

- A VPC and a vSwitch are created. For more information, see Create a VPC and Create a vSwitch in *VPC User Guide*.
- To associate a scaling group with SLB instances, make sure that the following requirements are met:

- You have one or more SLB instances in the **Running** state.
- The SLB instances and the scaling group are in the same organization, resource set, and region.

- To associate a scaling group with ApsaraDB RDS (RDS) instances, make sure that the following requirements are met:
  - You have one or more RDS instances in the **Running** state.
  - The RDS instances and the scaling group are in the same organization, resource set, and region.

Procedure

1.
2.
3.
4. In the upper-right corner of the Scaling Groups page, click **Create Scaling Group**.
5. Configure parameters for the scaling group.

Parameter	Required	Description
Scaling Group	Yes	The name of the scaling group. It must be 2 to 64 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or a digit.
Organization/Resource Group	Yes	The organization and resource set to which the scaling group belongs.
Maximum Capacity	Yes	The maximum number of instances that a scaling group can contain. Set the value based on business needs to control costs. Valid values: 0 to 1000.
Minimum Capacity	Yes	The minimum number of instances that a scaling group must contain. Set the value based on business needs to ensure service availability. When the scaling group is enabled, Auto Scaling automatically creates this number of ECS instances. Valid values: 0 to 1000.
Cooldown (Seconds)	Yes	The time during which Auto Scaling cannot execute any new scaling activities. The cooldown time occurs after the scaling group successfully executes a scaling activity. During the cooldown time, Auto Scaling rejects all scaling activity requests triggered by event-triggered tasks from Cloud Monitor. However, scaling activities triggered by other types of tasks such as manually triggered tasks and scheduled tasks are not limited by the cooldown time. These tasks are immediately executed. The value must be an integer that is greater than or equal to zero. Unit: seconds.

Parameter	Required	Description
Scale-In Policy	No	The policy for automatically removing ECS instances from the scaling group. This parameter contains the **Select** and **From the list, select** fields. You cannot specify the same values for the two fields. Valid values: ○ **Earliest Instance Created Using Scaling Configuration**: filters instances that were created based on the earliest scaling configuration. ○ **Earliest Created Instance**: filters instances that were added to the scaling group at the earliest point in time. ○ **Newest Instances**: filters instances that were most recently added to the scaling group. ○ **None**: is available only for **From the list, select**. This value indicates that Auto Scaling does not filter instances based on the From the list, select field. For example, if Auto Scaling filters instances based on the **Earliest Created Instance** value of Select, you can select one of the following values for From the list, select: ○ **None**: indicates that Auto Scaling does not filter instances based on the From the list, select field. ○ **Newest Instances**: filters instances obtained based on the Select field and then filters instances that were most recently added to the scaling group.
Region/VPC	Yes	The region and VPC to which the scaling group belongs.
VSwitch	Yes	The ID of the vSwitch with which the scaling group is associated.
Associate SLB Instance	No	After you associate SLB instances with the scaling group, ECS instances that are added to the scaling group are automatically added as SLB backend servers. You can specify a server group to which to add the ECS instances. ECS instances can be added to the following types of server groups: ○ Default server group: the group of ECS instances that are used to receive requests. If the listener is not configured with a vServer group or a primary/secondary server group, requests are forwarded to the ECS instances in the default server group. ○ vServer group: If you want to distribute different requests to different backend servers or configure domain name- or URL-based routing methods, you can use vServer groups.
Associate RDS Instance	No	After you associate RDS instances with the scaling group, the internal IP addresses of ECS instances that are added to the scaling group are automatically added to the whitelists of the RDS instances to allow internal communication.

6. Click **OK**.

Result

The created scaling group is displayed in the scaling group list but is in the **Disabled** state. To enable the scaling group, you must create a scaling configuration. For more information, see Create a scaling configuration.

4.1.4.2. Enable a scaling group

This topic describes how to enable a scaling group. You can enable a scaling group to trigger scaling activities.

Prerequisites

- The scaling group is in the **Disabled** state.
- The scaling group has a scaling configuration that is in the **Enable** state.

Procedure

1.
2.
3. Find the target scaling group and click **Enable** in the **Actions** column.
4. Click **OK**.

Result

In the **Status** column, the state of the scaling group is changed from **Disabled** to **Enable**.

4.1.4.3. View scaling groups

This topic describes how to view the scaling group list and the details of a specific scaling group.

Procedure

1.
2. In the top navigation bar, select an organization, a resource set, and a region.
   The scaling groups that correspond to the specified organization, resource set, and region are displayed.
3. Select a filter option, enter the corresponding information, and then click **Search**.

   You can select multiple filter options to narrow down the search results.

Option	Description
Scaling Group	Enter a scaling group name to search for the scaling group.
Scaling Group ID	Enter a scaling group ID to search for the scaling group.

4. Click the name of the scaling group in the **Scaling Group Name/ID** column.
5. View the details of the specified scaling group.

Parameter	Description
Basic Information	The configurations of the scaling group, such as the scaling group ID, scaling group name, total instances, minimum number of instances, maximum number of instances, and scale-in policy.
ECS Instances	The details of ECS instances, such as the list of automatically created ECS instances, the list of manually added ECS instances, and the number of ECS instances that are in service.

Parameter	Description
Scaling Activities	All the scaling activities that have been executed in the scaling group.
Scaling Configuration	The information of scaling configurations in the scaling group.
Scaling Rules	The information of scaling rules.

4.1.4.4. Modify a scaling group

This topic describes how to modify a scaling group. You can modify the parameters of a specific scaling group, such as the minimum and maximum numbers of ECS instances.

Context

After you modify the minimum or maximum number of ECS instances that a scaling group can have, if the number of instances in the scaling group is outside this range, Auto Scaling automatically creates or removes ECS instances until the number of instances are within the range.

Procedure

1.
2.
3. Find the target scaling group and click **Edit** in the **Actions** column.
4. Modify the parameters of the scaling group.

   You can modify the scaling configuration and other parameters, but not the organization and resource set. For more information about other parameters, see Create a scaling group.
5. Click **OK**.

4.1.4.5. Disable a scaling group

This topic describes how to disable a scaling group.

Prerequisites

- The scaling group does not have scaling activities in progress.
- The scaling group is in the **Enable** state.

Procedure

1.
2.
3. Find the target scaling group and click **Disable** in the **Actions** column.
4. Click **OK**.

Result

The status of the scaling group is changed from **Enable** to **Disabled** in the **Status** column.

4.1.4.6. Delete a scaling group

This topic describes how to delete a scaling group. When you delete a scaling group, Auto Scaling removes and releases ECS instances that are automatically created, removes ECS instances that are manually added, and deletes the scaling configurations and rules in the scaling group. However, the scheduled tasks and event-triggered tasks that are associated with the scaling group are not deleted.

Procedure

1.
2.
3. Find the target scaling group and click **Delete** in the **Actions** column.
4. Click **OK**.

4.1.4.7. Query ECS instances

You can query all ECS instances in a scaling group and their states.

Procedure

1.
2.
3.
4.
5.
6. View the details of ECS instances.

Category	Description
Automatically created ECS instances	The ECS instances that are automatically created based on the active scaling configuration when a scaling rule is triggered.
Manually added ECS instances	The ECS instances that are manually added to the specified scaling group.
The number of ECS instances in each state.	The following section describes the states: ○ Total: all ECS instances in the scaling group ○ In Service: the ECS instances that are in normal use ○ On Standby: the ECS instances that are on standby ○ Protected: the ECS instances that are protected ○ Adding: the ECS instances that are being added to the scaling group ○ Removing: the ECS instances that are being removed from the scaling group **Note** The Disabled, Adding:wait, and Suspending states are unavailable.

4.1.4.8. Put an ECS instance into the Standby state

This topic describes how to put an ECS instance into the Standby state. Auto Scaling does not perform health checks on or release ECS instances in the Standby state.

Context

After an ECS instance is put into the Standby state:

- The ECS instance stays in the Standby state until you change its status.
- Auto Scaling stops managing the lifecycle of the ECS instance. You must manually manage the lifecycle of the ECS instance.
- If a scale-in event is triggered, Auto Scaling will not remove the ECS instance.
- When the ECS instance is stopped or restarted, its health check status is not affected.
- To release the ECS instance, you must first remove it from the scaling group.
- If you delete the scaling group, the ECS instance is automatically put out of the Standby state and is released.
- You can also perform other operations on the ECS instance, such as stopping, restarting, changing the instance type of, or changing the operating system of the ECS instance.

Procedure

1.
2.
3.
4.
5.
6. Find the target ECS instance and choose **Actions > Switch to Standby** in the **Actions** column.
7. Click **OK**.

4.1.4.9. Remove an ECS instance from the Standby state

This topic describes how to remove an ECS instance from the Standby state. You can remove an instance from the Standby state to reuse it.

Context

After an ECS instance is removed from the Standby state:

- The ECS instance enters the In Service state.
- When the ECS instance is stopped or restarted, its health status is updated.
- Auto Scaling continues to manage the lifecycle of the ECS instance, and can remove the ECS instance from the scaling group during a scale-in event.

Procedure

1.
2.
3.
4.
5.
6. Find the target ECS instance and choose **Actions > Move Out Of Standby** in the **Actions** column.
7. Click **OK**.

4.1.4.10. Put an ECS instance into the Protected state

This topic describes how to put an ECS instance into the Protected state. Auto Scaling does not perform health checks on or release ECS instances that are in the Protected state.

Context

After an ECS instance is put into the Protected state:

- The ECS instance stays in the Protected state until you change its status.
- If a scale-in event is triggered, Auto Scaling will not remove the ECS instance. To release the ECS instance, you must remove the ECS instance from the Protected state and then remove it from the scaling group.
- When the ECS instance is stopped or restarted, its health check status is not affected.

Procedure

1.
2.
3.
4.
5.
6. Find the target ECS instance and choose **Actions > Switch to Protection** in the **Actions** column.
7. Click **OK**.

4.1.4.11. Remove an ECS instance from the Protected state

This topic describes how to remove an ECS instance from the Protected state. After an ECS instance is removed from the Protected state, Auto Scaling continues to manage the lifecycle of the ECS instance.

Procedure

1.
2.
3.
4.
5.
6. Find the target ECS instance and choose **Actions > Move Out Of Protection** in the **Actions** column.
7. Click **OK**.

4.1.5. Scaling configurations

4.1.5.1. Create a scaling configuration

This topic describes how to create a scaling configuration for a scaling group.

Prerequisites

At least one security group is available. If you do not have any security groups, create a security group. For more information, see Create a security group *in ECS User Guide.*

Context

You can create only a limited number of scaling configurations for a scaling group. For more information, see the Limits topic in *Auto Scaling Product Introduction.*

Procedure

1.
2.

3.
4.
5. Choose **Create > Create Scaling Configuration**.
6. Configure parameters for the scaling configuration.

Section	Parameter	Required	Description
Region	Region	Yes	The region where the ECS instance is located.
	Zone	Yes	The zone where the ECS instance is located.
Security Group	Security Group	Yes	The security group to which the ECS instance belongs.
Instance	Instance Family	Yes	The instance family to which the ECS instance belongs.
	Instance Type	Yes	The instance type of the ECS instance.
Image	Image Type	Yes	◦ **Public Image**: Public images provided by Alibaba Cloud are fully licensed to offer a secure and stable operating environment for applications on ECS instances. ◦ **Custom Image**: You can create custom images to install software or deploy projects that have special requirements.
Storage	System Disk	Yes	Specify the category and size of the system disk. The operating system is installed on the system disk. You can select **Ultra Disk** or **Standard SSD**.
	Data Disk	No	Specify the category and size of the data disk. You can select **Ultra Disk** or **Standard SSD**. You can add a maximum of 16 data disks. The maximum capacity of each data disk is 32 TiB. You can set **Release with Instance** and **Encrypt** for each data disk.
	Set Password	Yes	Select when to set password. You can select **Now** or **Later**. If you select Later, you can use the Change Password feature in the console to set the password. For more information, see the Change Password topic in *ECS User Guide.*

Section	Parameter	Required	Description
Password	Logon Password	No	The password used to log on to the ECS instance. The password must be 8 to 30 characters in length and must contain at least three of the following character types: digits, uppercase letters, lowercase letters, and special characters. **Note** The password is used to log on to the operating system and is not the VNC password.
	Confirm Password	No	Enter the password again.
Deployment Set	Deployment Set	No	The deployment set to which the instance belongs.
Instance Name	Configuration Name	No	The name of the scaling configuration.
	Instance Name	No	The name of the ECS instance.
User Data	User Data	No	Windows supports two formats: Bat and Powershell. Before you perform Base64 encoding, make sure to include `[bat]` or `[powershell]` as the first line. You can run shell scripts for Linux ECS instances.
Quantity	Quantity	No	The number of instances to purchase.

7. Click **Submit**.

Result

After the scaling configuration is created, it is in the **Disabled** state and is displayed in your scaling configuration list. To automatically create ECS instances, you must apply a scaling configuration. For more information, see Apply a scaling configuration.

4.1.5.2. View scaling configurations

This topic describes how to view scaling configurations.

Procedure

1.
2. In the top navigation bar, select an organization, a resource set, and a region.
   The scaling groups that correspond to the specified organization, resource set, and region are displayed.
3.
4.
5. View the list of scaling configurations.

4.1.5.3. Modify a scaling configuration

This topic describes how to modify a scaling configuration. You can modify the parameters of a scaling configuration based on your actual needs.

Procedure

1.
2.
3.
4.
5. Find the target scaling configuration and click its name in the **Scaling Configuration Name/ID** column.
6. Modify the parameters of the scaling configuration.

   For more information about parameters of the scaling configuration, see Create a scaling configuration.
7. Click **OK**.

4.1.5.4. Apply a scaling configuration

This topic describes how to apply a scaling configuration. You can create multiple scaling configurations for a scaling group and apply one.

Procedure

1.
2.
3.
4.
5. Find the target scaling configuration and click **Select** in the **Actions** column.

   Only one scaling configuration can be in the **Enabled** state in a scaling group. After a scaling configuration is applied, other scaling configurations are put into the **Disabled** state.
6. Click **OK**.

Result

The status of the scaling configuration changes from **Disabled** to **Enable** in the **Status** column.

4.1.5.5. Delete a scaling configuration

This topic describes how to delete a scaling configuration that is no longer needed. After you delete a scaling configuration, existing ECS instances that are created from the scaling configuration are not removed.

Prerequisites

The scaling configuration is in the **Disabled** state.

Procedure

1.
2.
3.
4.
5. Find the target scaling configuration and click **Delete** in the **Actions** column.

6. Click **OK**.

4.1.6. Scaling rules

4.1.6.1. Create a scaling rule

This topic describes how to create a scaling rule. You can create scaling rules to add or remove ECS instances. For example, you can add an ECS instance to a scaling group.

Context

- You can create only a limited number of scaling rules for a scaling group. For more information, see the Limits topic in *Auto Scaling Product Introduction.*
- After a scaling rule is executed, the resulting number of ECS instances in the scaling group may fall outside of the specified range. In this case, Auto Scaling automatically adjusts the number of ECS instances to ensure that the number of ECS instances in the scaling group is within the specified range.

Procedure

1.
2.
3.
4.
5. Click **Create Scaling Rule**.
6. Configure parameters for the scaling rule.

Parameter	Required	Description
Rule Name	Yes	The name of the scaling rule. It must be 2 to 64 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.
Scaling Activity	Yes	The operation that is performed when the scaling rule is triggered. The operations include: ○ **Change to N instances**: After the scaling rule is executed, the number of instances in the scaling group is changed to N. ○ **Add N instances**: After the scaling rule is executed, N instances are added to the scaling group. ○ **Remove N instances**: After the scaling rule is executed, N instances are removed from the scaling group.
Default Cooldown (Seconds)	No	The cooldown period. If this parameter is not specified, the default value is used.

7. Click **OK**.

4.1.6.2. View scaling rules

This topic describes how to view scaling rules.

Procedure

1.
2. In the top navigation bar, select an organization, a resource set, and a region.
   The scaling groups that correspond to the specified organization, resource set, and region are displayed.
3.
4.
5. View the list of scaling rules.

4.1.6.3. Modify a scaling rule

This topic describes how to modify a scaling rule. You can modify the following parameters of a scaling rule: Rule Name, Scaling Activity, and Default Cooldown.

Procedure

1.
2.
3.
4.
5. Find the target scaling rule and click **Edit** in the **Actions** column.
6. Modify the Rule Name, Scaling Activity, and Default Cooldown parameters.
7. Click **OK**.

4.1.6.4. Delete a scaling rule

This topic describes how to delete a scaling rule that is no longer needed.

Procedure

1.
2.
3.
4.
5. Find the target scaling rule and click **Delete** in the **Actions** column.
6. In the message that appears, click **OK**.

4.1.7. Scaling tasks

4.1.7.1. Manually execute a scaling rule

This topic describes how to manually execute a scaling rule to add or remove ECS instances.

Prerequisites

- The scaling group to which the scaling rule belongs is in the **Enable** state.
- No scaling activity is in progress in the scaling group to which the scaling rule belongs.

Context

After the scaling rule is executed, if the number of ECS instances is greater than the maximum number or less than the minimum number, Auto Scaling automatically adjusts the number of ECS instances to be within the valid range.

Auto Scaling enables you to manually execute scaling rules. You can also associate an event-triggered task or scheduled task with the scaling rule to automatically adjust the number of ECS instances. For more information, see Create a scheduled task and Create an event-triggered task.

Procedure

1.
2.
3.
4.
5. Find the scaling rule that you want to execute and click **Run** in the **Actions** column.
6. In the message that appears, click **OK**.

Result

The **Scaling Activities** page appears. You can view the details of your scaling activity.

4.1.7.2. Manually add an ECS instance

This topic describes how to manually add an ECS instance to a scaling group. You can add existing ECS instances to a scaling group to take full advantage of the computing resources.

Prerequisites

The ECS instance to be added must meet the following conditions:

- The ECS instance and the scaling group to which to add the instance share the same region, organization, and resource set.
- The ECS instance is in the **Running** state.
- The ECS instance does not belong to any scaling groups.
- The ECS instance and the scaling group are in the same VPC.

The scaling group to which to add the ECS instance must meet the following conditions:

- The scaling group is in the **Enable** state.
- No scaling activity is in progress in the scaling group.

Context

- When no scaling activity is being executed in the scaling group, you can add an ECS instance to the scaling group without the need to wait for the cooldown time to expire.
- If the number of instances in the scaling group is greater than the maximum number of instances after an ECS instance is added to the scaling group, the ECS instance cannot be added.
- The ECS instances that are manually added to a scaling group are not limited by scaling configurations. The instance types of the manually added instances can be different from that of the scaling configuration in the **Enable** state.

Procedure

1.
2.
3.
4.
5. Click **Add Instance**.
6. Select the ECS instance to be added and click **OK**.

Result

The manually added instance is displayed on the **Manually Added** tab.

4.1.7.3. Manually remove an ECS instance

This topic describes how to manually remove an ECS instance that is no longer needed from a scaling group.

Prerequisites

The scaling group must meet the following conditions:

- The scaling group is in the **Enable** state.
- No scaling activity is in progress in the scaling group.

Context

- When no scaling activity is being executed in the scaling group, you can immediately remove an ECS instance from the scaling group without the need to wait for the cooldown time to expire.
- After an ECS instances is removed from a scaling group, the number of instances in the scaling group must be greater than or equal to the minimum number of instances. Otherwise, the ECS instance cannot be removed.

Procedure

1.
2.
3.
4.
5.
6. Use one of the following methods to remove one or more ECS instances from a scaling group:

   Manually added ECS instances can only be removed, but cannot be released.

   - Find the ECS instance that you want to remove and choose **Actions > Remove from Scaling Group** in the **Actions** column.
   - Find the ECS instance that you want to remove and release, and choose **Actions > Remove from Scaling Group and Release** in the **Actions** column.
7. In the message that appears, click **OK**.

4.1.8. Scheduled tasks

4.1.8.1. Create a scheduled task

This topic describes how to create a scheduled task to scale computing resources in response to predictable business changes in the future. Scheduled tasks enable the system to automatically obtain sufficient computing resources before business peaks and release idle computing resources after the business peaks.

Context

A scheduled task is preconfigured to execute the specified scaling rule at the specified time. When the specified time arrives, the scheduled task automatically scales computing resources. This allows you to reduce costs and meet business requirements. You can also specify the recurrence for scheduled tasks if business changes are regular.

If multiple scheduled tasks need to be executed in 1 minute, Auto Scaling executes the most recently created scheduled task.

Procedure

1.
2.
3.
4. In the upper-right corner of the Scheduled Tasks page, click **Create Scheduled Task**.
5. In the dialog box that appears, configure parameters for the scheduled task.

Parameter	Required	Description
Task Name	Yes	The name of the scheduled task. The name must be 2 to 64 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.
Description	Yes	The description of the scheduled task.
Organization/Resource Group	Yes	The organization and resource set to which the scheduled task belongs.
Start Time	Yes	The time to execute the scheduled task.
Scaling Rules	Yes	The scaling group to be monitored and the scaling rule to be executed.
Retry Interval (Seconds)	No	The period of time during which the system retires to execute the scheduled task. Unit: seconds. If a scaling activity fails to be executed at the specified time, Auto Scaling executes the scheduled task again within the period of time that is specified by the Retry Interval (Seconds) parameter.
Recurrence Settings (Advanced)	No	This parameter specifies whether to repeatedly execute the scheduled task. Select **Recurrence Settings (Advanced)** and set the Recurrence and Expire parameters. The recurrence values include **Daily**, **Weekly**, and **Monthly**.

6. Click **OK**.

Result

The scheduled task that you created is displayed in the scheduled task list.

4.1.8.2. View scheduled tasks

This topic describes how to view scheduled tasks.

Procedure

1.
2.
3. In the top navigation bar, select an organization, a resource set, and a region.
   The scheduled tasks that correspond to the specified organization, resource set, and region are displayed.
4. Select a filter option, enter the corresponding information, and then click **Search**.

   You can select multiple filter options to narrow down the search results.

Option	Description
Task Name	Enter a task name to search for the scheduled task.

Option	Description
Task ID	Enter a task ID to search for the scheduled task.

5. View the scheduled task list.

4.1.8.3. Modify a scheduled task

This topic describes how to modify a scheduled task. You can modify parameters such as Start Time, Scaling Rules, and Retry Expiry Time for a scheduled task.

Procedure

1.
2.
3.
4. Find the scheduled task that you want to modify and click **Edit** in the **Actions** column.
5. Modify the parameters of the scheduled task.

   You can modify the Recurrence and Expire parameters if you have enabled the Recurrence Settings (Advanced) feature when you create the scheduled task, but the Recurrence Settings (Advanced) feature cannot be disabled. For more information about other parameters of the scheduled task, see Create a scheduled task.
6. Click **OK**.

4.1.8.4. Disable a scheduled task

This topic describes how to disable a scheduled task. You can disable a scheduled task that is no longer needed.

Prerequisites

The scheduled task is in the **Running** state.

Procedure

1.
2.
3.
4. Find the scheduled task that you want to disable and click **Disabled** in the **Actions** column.
5. In the message that appears, click **OK**.

Result

The status of the scheduled task is changed from **Running** to **Stop** in the **Status** column.

4.1.8.5. Enable a scheduled task

This topic describes how to enable a scheduled task. You can enable a scheduled task that has been disabled and use it to trigger scaling activities at the specified time point.

Prerequisites

The scheduled task is in the **Stop** state.

Procedure

1.
2.
3.
4. Find the scheduled task that you want to enable and click **Enable** in the **Actions** column.
5. In the message that appears, click **OK**.

Result

The status of the scheduled task is changed from **Stop** to **Running** in the **Status** column.

4.1.8.6. Delete a scheduled task

This topic describes how to delete a scheduled task that is no longer needed.

Procedure

1.
2.
3.
4. Find the scheduled task that you want to delete and click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

4.1.9. Event-triggered tasks

4.1.9.1. Create an event-triggered task

This topic describes how to create an event-triggered task associated with monitoring metrics in response to emergent or unpredictable business changes. After you create and enable an event-triggered task, Auto Scaling collects data for the specified metric in real time and triggers an alert when the specified condition is met. Then, Auto Scaling executes the scaling rule to scale ECS instances in the scaling group.

Procedure

1.
2.
3.
4. In the upper-right corner of the Event-Triggered Tasks page, click **Create Alert**.
5. In the dialog box that appears, configure parameters for the event-triggered task.

Parameter	Required	Description
Task Name	Yes	The name of the event-triggered task. It must be 2 to 64 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or a digit.
Description	No	The description of the event-triggered task.
Organization/Resource Group	Yes	The organization and resource set in which to create the event-triggered task.
Monitoring Metrics/Scaling Rules	Yes	The scaling group to be monitored and the scaling rule to be executed.

Parameter	Required	Description
Monitoring Type	Yes	**System-Level Monitoring** is selected by default.
Monitoring Metrics	Yes	The metrics that you want to monitor. Valid values: ○ **Average CPU Utilization** ○ **Memory Usage** ○ **Outbound Traffic** ○ **Inbound Traffic**
Monitoring Period	Yes	The period during which data is aggregated and analyzed. The shorter the period, the higher the frequency that the alert is triggered. Unit: minutes. Valid values: ○ 1 ○ 2 ○ 5 ○ 15
Statistic	Yes	The rule that determines whether to trigger an alert. Select **Average**, **Max Capacity**, or **Min Capacity**, and specify a threshold value. For example, to trigger an alert when the CPU utilization exceeds 80%: ○ Average: An alert is triggered when the average CPU utilization of all ECS instances in the scaling group exceeds 80%. ○ Max Capacity: An alert is triggered when the highest CPU utilization among the ECS instances in the scaling group exceeds 80%. ○ Min Capacity: An alert is triggered when the lowest CPU utilization among the ECS instances in the scaling group exceeds 80%.
Trigger After	Yes	The number of consecutive times that the threshold must be exceeded before the alert is triggered. Valid values: ○ 1 ○ 2 ○ 3 ○ 5

6. Click **OK**.

4.1.9.2. View event-triggered tasks

This topic describes how to view event-triggered tasks.

Procedure

1.
2.
3. In the top navigation bar, select an organization, a resource set, and a region.
   The event-triggered tasks that correspond to the specific organization, resource set, and region are displayed.

4. Select a filter option, enter the corresponding information, and then click **Search**.

   You can select multiple filter options to narrow down the search results.

Option	Description
Alert Name	Enter an event-triggered task name to search for the event-triggered task.
Scaling Group ID	Enter a scaling group ID to search for the event-triggered task associated with the scaling group.

4.1.9.3. Modify an event-triggered task

This topic describes how to modify an event-triggered task. You can modify parameters such as Scaling Rules, Monitoring Type, and Statistic for an event-triggered task.

Procedure

1.
2.
3.
4. Find the event-triggered task that you want to modify and click **Edit** in the **Actions** column.
5. Modify the parameters of the event-triggered task.

   For more information about other parameters of the scheduled task, see Create an event-triggered task. The following parameters cannot be modified:

   - Organization
   - Resource Group
   - Monitoring Metrics
   - Monitoring Period
6. Click **OK**.

4.1.9.4. Disable an event-triggered task

This topic describes how to disable an event-triggered task. You can disable an event-triggered task if you no longer want to use it to trigger scaling activities.

Prerequisites

The event-triggered task is in the **Normal**, **Alerts**, or **Insufficient Data** state.

Procedure

1.
2.
3.
4. Find the event-triggered task that you want to disable and click **Disable** in the **Actions** column.
5. In the message that appears, click **OK**.

Result

The status of the event-triggered task is changed to **Stopped** in the **Status** column.

4.1.9.5. Enable an event-triggered task

This topic describes how to enable an event-triggered task. You can enable an event-triggered task that has been disabled to continue to monitor metrics and trigger scaling activities for a scaling group.

Prerequisites

The event-triggered task is in the **Stopped** state.

Procedure

1.
2.
3.
4. Find the event-triggered task that you want to enable and click **Enable** in the **Actions** column.
5. In the message that appears, click **OK**.

Result

The status of the event-triggered task changes from **Stopped** to **Normal** in the **Status** column.

4.1.9.6. Delete an event-triggered task

This topic describes how to delete an event-triggered task. You can delete an event-triggered task that is no longer needed.

Procedure

1.
2.
3.
4. Find the event-triggered task that you want to delete and click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

5.Resource Orchestration Service (ROS)

5.1. User Guide

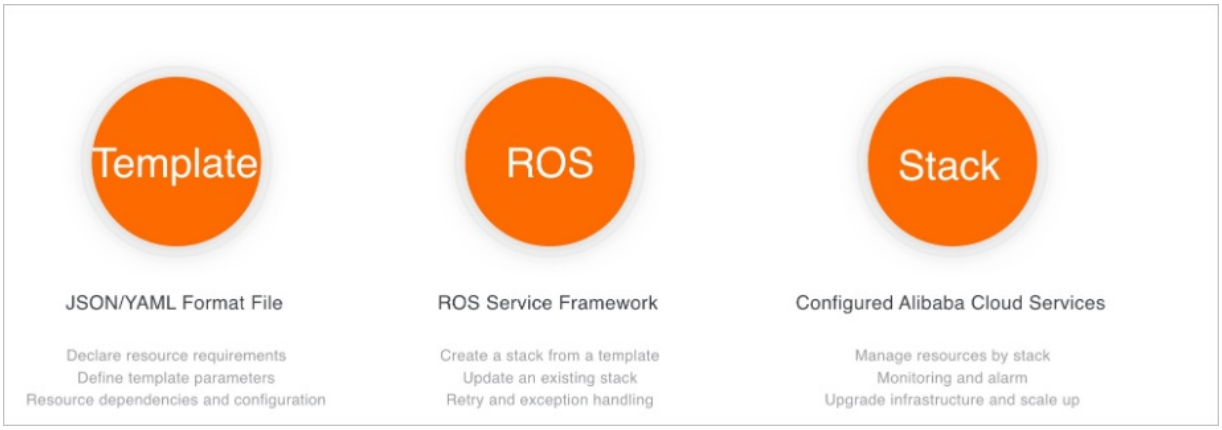
5.1.1. What is ROS?

Resource Orchestration Service (ROS) is a service provided by Alibaba Cloud to simplify the management of cloud computing resources. You can author stack templates based on the template specifications defined in ROS. Within a template, you can define required cloud computing resources such as ECS and ApsaraDB RDS instances, and the dependencies between resources. The ROS engine automatically creates and configures all resources in a stack based on a template, which makes automatic deployment and O&M possible.

An ROS template is a readable, easy-to-author text file. You can directly edit a JSON-formatted template or use the Visual Editor available in the ROS console to edit the template. You can modify templates at any time. You can use version control tools such as SVN and Git to control the template and infrastructure versions. You can use APIs and SDKs to integrate the orchestration capabilities of ROS with your own applications to implement Infrastructure as Code (IaC).

ROS templates are also a standardized way to deliver resources and applications. If you are an independent software vendor (ISV), you can use ROS templates to deliver a holistic system and solution encompassing cloud resources and applications. ISVs can use this method to integrate Alibaba Cloud resources with their own software systems for centralized delivery.

ROS manages a group of cloud resources as a single unit called a stack. A stack is a group of Alibaba Cloud resources. You can create, delete, and clone cloud resources by stack. In DevOps practices, you can use ROS to clone the development, test, and production environments to simplify the overall migration and scaling of applications.



5.1.2. Log on to the ROS console

This topic describes how to log on to the ROS console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password that you can use to log on to the console from the operations administrator.

> **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
>
> - Uppercase or lowercase letters
> - Digits
> - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products > Elastic Computing > Resource Orchestration Service**.

5.1.3. Create a stack

This topic describes how to create a stack in the Resource Orchestration Service (ROS) console.

Prerequisites

Log on to the ROS console. For more information, see Log on to the ROS console.

Procedure

1. In the upper-right corner of the page, click **Create Stack**.
2. In the **Select Template** step, set **Organization**, **Resource Set**, and **regionId**.
3. In the **Prepare Template** section, enter template content in the JSON format. Click **Next**.
4. In the **Configure Template Parameters** step, enter the stack name and parameters, and click **Next**.
5. In the **Configure Stack** step, set Rollback on Failure and Timeout Period, and click **Next**.
6. In the **Confirm** step, check the configurations of template and stack, and click **Create Stack**.

What's next

On the **Stacks** page in the ROS console:

- To delete a stack, click **Delete** in the Actions column corresponding to the stack.
- To update a stack, click **Update** in the Actions column corresponding to the stack.
- To recreate a stack, click **Recreate** in the Actions column corresponding to the stack.

> **Note**
>
> - If you only need to modify the current template and configurations of a specified stack but do not need to change the region where the stack resides, update the stack.
> - If you need to modify the current template and configurations of a specified stack and change the region where the stack resides, recreate the stack.

5.1.4. Template syntax

5.1.4.1. Template structure

A template is a UTF-8 encoded JSON file that is used to create stacks. Templates serve as the blueprint for underlying infrastructure and architecture. Templates define the configurations and dependencies of Alibaba Cloud resources.

ROS template structure

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Description" : "The template description used to provide information such as application scenarios and stack architecture.",
  "Metadata" : {
    // The template metadata that provides information such as layout for visualizations.
  },
  "Parameters" : {
    // The parameters you can specify when you create a stack.
  },
  "Mappings" : {
    // The mapping tables. Mapping tables are nested tables.
  },
  "Conditions": {
    // The conditions defined using internal condition functions. These conditions determine when to create associated resources.
  },
  "Resources" : {
    // The detailed information of resources such as configurations and dependencies.
  },
  "Outputs" : {
    // The outputs used to provide information such as resource properties. You can use the ROS console or API to obtain the information.
  }
}
```

ROSTemplateFormatVersion

Required. The template versions supported by Resource Orchestration Service (ROS). Current version: 2015-09-01.

Description

Optional. The description of the template, which is used to provide information such as the application scenarios and architecture of the template.

A detailed description can help users better understand the content of the template.

Metadata

Optional. The metadata of the template, in the JSON format.

Parameters

Optional. The parameters that you can specify when you create a stack. An ECS instance type is often defined as a parameter. Parameters have default values. Parameters can improve the flexibility and reusability of the template. When you create a stack, select appropriate specifications.

Mappings

Optional. Mappings are defined as nested mapping tables. You can use Fn::FindInMap to retrieve values corresponding to keys. You can also use parameter values as keys. For example, you can search the region-image mapping table for desired images by region.

Conditions

Optional. The conditions defined using Fn::And, Fn::Or, Fn::Not, and Fn::Equals. Separate multiple conditions with commas (,). The system will evaluate all conditions in the template before creating or updating a stack. All resources associated with true conditions are created, and all resources associated with false conditions are ignored.

Resources

Optional. The detailed information of resources in the stack created based on the template. The information includes resource dependencies and configurations.

Outputs

Optional. The outputs that are used to provide information such as resource properties. You can use the ROS console or API to obtain the information.

5.1.4.2. Parameters

The Parameters section improves the flexibility and reusability of a template. When you create a stack, you can replace parameter values in the template.

For example, assume that you have a web application requiring a stack that contains one SLB instance, two ECS instances, and one ApsaraDB RDS instance. If the web application has a heavy workload, you can select an ECS instance with high specifications when you create the stack. Otherwise, you can select an ECS instance with low specifications. The following example shows how to define the InstanceType parameter for an ECS instance:

```
"Parameters" : {
  "InstanceType" : {
    "Type" : "String",
    "AllowedValues":["ecs.t1.small","ecs.s1.medium", "ecs.m1.medium", "ecs.c1.large"],
    "Default": "ecs.t1.small",
    "Label": "The ECS instance type",
    "Description" : "The type of the ECS instance you want to create. Default value: ecs.t1.small. Valid values: ecs.t1.small, ecs.s1.medium, ecs.m1.medium, and ecs.c1.large."
  }
}
```

You can assign a value to the InstanceType parameter when you create stacks based on templates. If this parameter is not specified, the default value `ecs.t1.small` is used.

The following example shows how to reference the InstanceType parameter when you define a resource:

```
"Webserver" : {
  "Type" : "ALIYUN::ECS::Instance",
  "InstanceType": {
    "Ref": "InstanceType"
  }
}
```

Syntax

Each parameter consists of a name and properties. The parameter name can contain only letters and digits and must be unique within the template. You can use the Label field to define the alias of the parameter.

The following table describes the parameter properties.

Parameter property	Required	Description

Parameter property	Required	Description
Type	Yes	The data type of the parameter. • String: a string value. Example: `"ecs.s1.medium"`. • Number: an integer or floating-point number. Example: 3.14. • CommaDelimitedList: a set of strings or numbers separated by commas (,), which can be indexed by using the Fn::Select function. Example: `"80, foo, bar"`. • Json: a JSON string. Example: `{ "foo": "bar" }`. • Boolean: a Boolean value. Example: `true` or `false`.
Default	No	If you do not specify a value when you create a stack, Resource Orchestration Service (ROS) checks whether a default value is defined in the template. If a default value is found, ROS uses the default value. Otherwise, an error is returned.
AllowedValues	No	The list of one or more valid parameter values.
AllowedPattern	No	The regular expression that is used to check whether the specified parameter value is a string. If the input is not a string, an error is returned.
MaxLength	No	The integer value that determines the longest string allowed for a String-type parameter.
MinLength	No	The integer value that determines the shortest string allowed for a String-type parameter.
MaxValue	No	The numeric value that determines the maximum value allowed for a Number-type parameter.
MinValue	No	The numeric value that determines the minimum value allowed for a Number-type parameter.
NoEcho	No	Specifies whether to mask the parameter value when the GetStack operation is called. If you set this parameter property to true, only asterisks (*) are returned.
Description	No	The string that describes the parameter.
ConstraintDescription	No	The description of the parameter constraints.
Label	No	The alias of the parameter, encoded in UTF-8. When you create a web form based on a template, the Label value can be mapped to the parameter name.

Parameter property	Required	Description
AssociationProperty	No	The associated resource property. If you specify this parameter property, ROS verifies whether the specified parameter value is valid and provides a list of valid values based on the associated resource property. Valid values: • `ALIYUN::ECS::Instance::ImageId` • `ALIYUN::ECS::Instance::ZoneId` • `ALIYUN::ECS::VPC::VPCId` • `ALIYUN::ECS::VSwitch::VSwitchId` For example, if you set `AssociationProperty` to `ALIYUN::ECS::Instance::ImageId`, ROS verifies whether the specified image ID is valid and lists other valid values in a drop-down list.
Confirm	No	Specifies whether to enter the parameter value for a second time if the NoEcho property is set to true. Default value: false. **Notice** The Confirm property can be set to true only when it is used together with a String-type parameter and when the NoEcho property is set to true.

Examples

In the following example, two parameters are defined in the Parameters section.

- `username`
  - Type: String
  - Default value: anonymous. Valid values:
    - anonymous
    - user-one
    - user-two
  - Length: 6 to 12

  Notice The default value must also meet the length and valid value requirements.
- `password`
  - Type: String
  - Length: 1 to 41
  - The password can contain uppercase letters, lowercase letters, and digits.
  - If you set the NoEcho property to true, the GetStack operation does not return any parameter values.

```
"Parameters" : {
"username" : {
"Label": "Username",
"Description" : "Enter the username",
"Default": "anonymous",
"Type" : "String",
"MinLength" : "6",
"MaxLength" : "12",
"AllowedValues": ["anonymous", "user-one", "user-two"]
},
"password" : {
"Label": "Password",
"NoEcho" : "True",
"Description" : "Enter the password",
"Type" : "String",
"MinLength" : "1",
"MaxLength" : "41",
"AllowedPattern" : "[a-zA-Z0-9]*"
}
}
```

Pseudo parameters

Pseudo parameters are internal parameters provided by the ROS engine. They can be referenced in the same manner as user-defined parameters, and their values are determined when ROS is running. The following pseudo parameters are supported:

- ALIYUN::StackName: the name of the stack.
- ALIYUN::StackId: the ID of the stack.
- ALIYUN::Region: the region where the stack resides.
- ALIYUN::AccountId: the user ID of the stack.
- ALIYUN::NoValue: specifies whether the specific resource property is deleted when the resource is created or updated.

5.1.4.3. Resources

This topic describes the properties of each resource and dependencies of resources in a stack. A resource can be referenced by other resources and output items.

Syntax

Each resource consists of an ID and a description. All resource descriptions are enclosed in braces {}. Separate multiple resources with commas (,). The following sample code shows the Resources syntax:

```
"Resources" : {
    "Resource1 ID": {
        "Type": "The resource type",
        "Condition": "The condition that specifies whether to create the resource",
        "Properties" : {
            The description of the resource properties
        }
    },
    "Resource2 ID" : {
        "Type": "The resource type",
        "Condition": "The condition that specifies whether to create the resource",
        "Properties" : {
            The description of the resource properties
        }
    }
}
```

Parameter description:

- The resource ID must be unique within the template. You can use the resource ID to reference the resource in other parts of the template.
- The Type parameter specifies the type of resource that is being declared. For example, ALIYUN::ECS::Instance indicates that the resource is an Elastic Cloud Service (ECS) instance.
- The Properties section provides additional options that you can specify for a resource. For example, you must specify an image ID for each Alibaba Cloud ECS instance. The image ID is one of the resource properties.

Examples

```
"Resources" : {
  "ECSInstance" : {
    "Type" : "ALIYUN::ECS::Instance",
    "Properties" : {
      "ImageId" : "m-25l0r****"
    }
  }
}
```

If a resource does not need properties to be declared, omit the Properties section of that resource.

Property values can be text strings, string lists, Boolean values, referenced parameters, or return values of functions.

The following example shows how to declare different types of property values:

```
"Properties" : {
    "String" : "string",
    "LiteralList" : [ "value1", "value2" ],
    "Boolean" : "true"
    "ReferenceForOneValue" :  { "Ref" : "ResourceID" } ,
    "FunctionResultWithFunctionParams" : {
        "Fn::Join" : [ "%", [ "Key=", { "Ref" : "SomeParameter" } ] ] }
}
```

DeletionPolicy

The DeletionPolicy parameter specifies whether to retain a resource when its stack is deleted. The following sample code shows how to use the DeletionPolicy parameter to retain an ECS instance when its stack is deleted:

```
"Resources" : {
  "ECSInstance" : {
    "Type" : "ALIYUN::ECS::Instance",
    "Properties" : {
      "ImageId" : "m-25l0r****"
    },
    "DeletionPolicy" : "Retain"
  }
}
```

DependsOn

The DependsOn parameter allows you to create a specific resource after you create its dependent resource. If you specify the DependsOn parameter for a resource, the resource is created only after its dependent resource specified by the DependsOn parameter is created.

In the following example, WebServer is created only after DatabaseServer is created:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Resources" : {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "DependsOn": "DatabseServer"
    },
    "DatabseServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId" : "m-25l0r****",
        "InstanceType": "ecs.t1.small"
      }
    }
  }
}
```

Condition

The Condition parameter specifies whether to create the resource. The resource can be created only when the Condition parameter is set to true.

In the following example, WebServer is created only if the condition determined by the MaxAmount parameter is true:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Parameters": {
    "MaxAmount": {
      "Type": "Number",
      "Default": 1
    }
  },
  "Conditions": {
    "CreateWebServer": {"Fn::Not": {"Fn::Equals": [0, {"Ref": "MaxAmount"}]}}
  }
  "Resources" : {
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceGroup",
      "Condition": "CreateWebServer",
      "Properties": {
        "ImageId" : "m-25l0r****",
        "InstanceType": "ecs.t1.small"
        "MaxAmount": {"Ref": "MaxAmount"}
      }
    },
    "DatabseServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId" : "m-25l0r****",
        "InstanceType": "ecs.t1.small"
      }
    }
  }
}
```

Resource declaration example

The following example shows how to declare a resource:

```
"Resources" : {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId" : "m-25l0r****",
        "InstanceType": "ecs.t1.small",
        "SecurityGroupId": "sg-25zwc****",
        "ZoneId": "cn-beijing-b",
        "Tags": [{
            "Key": "Department1",
            "Value": "HumanResource"
        },{
            "Key": "Department2",
            "Value": "Finance"
        }
        ]
      }
    },
    "ScalingConfiguration": {
      "Type": "ALIYUN::ESS::ScalingConfiguration",
      "Properties": {
        "ImageId": "ubuntu1404_64_20G_aliaegis_2015****.vhd",
        "InstanceType": "ecs.t1.small",
        "InstanceId": "i-25xhh****",
        "InternetChargeType": "PayByTraffic",
        "InternetMaxBandwidthIn": 1,
        "InternetMaxBandwidthOut": 20,
        "SystemDisk_Category": "cloud",
        "ScalingGroupId": "bwhtvpcBcKYac9fe3vd0****",
        "SecurityGroupId": "sg-25zwc****",
        "DiskMappings": [
            {
                "Size": 10
            },
            {
                "Category": "cloud",
                "Size": 10
            }
        ]
      }
    }
}
```

5.1.4.4. Outputs

The Outputs section is used to define the values returned when the GetStack operation is called. For example, if you define an ECS instance ID as an output item, the ECS instance ID is returned when the GetStack operation is called.

Syntax

Each output item consists of an ID and a description. All output descriptions are enclosed in braces {}. Separate multiple output items with commas (,). Each output item can have multiple values in the array format. The following example shows the Outputs syntax:

```
"Outputs" : {
  "Output1 ID" : {
    "Description": "The description of the output item",
    "Condition": "The condition that specifies whether to provide resource properties",
    "Value": "The output value expression"
  },
  "Output2 ID" : {
    "Description": "The description of the output item",
    "Condition": "The condition that specifies whether to provide resource properties",
    "Value" : [
      "Output value expression 1",
      "Output value expression 2",
      ...
    ]
  }
}
```

- Output ID: the ID of the output item. Duplicate IDs are not allowed within a template.
- Description: optional. The description of the output item.
- Value: required. The value returned when the GetStack operation is called.
- Condition: optional. The condition that specifies whether to create a resource and provide its information. The resource is created and its information is provided only when the specified condition is true.

  In the following example, WebServer is created only if the condition determined by the MaxAmount parameter is true:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Parameters": {
    "MaxAmount": {
      "Type": "Number",
      "Default": 1
    }
  },
  "Conditions": {
    "CreateWebServer": {"Fn::Not": {"Fn::Equals": [0, {"Ref": "MaxAmount"}]}}
  }
  "Resources" : {
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceGroup",
      "Condition": "CreateWebServer",
      "Properties": {
        "ImageId" : "m-25l0r****",
        "InstanceType": "ecs.t1.small"
        "MaxAmount": {"Ref": "MaxAmount"}
      }
    }
  }
  "Outputs": {
    "WebServerIP": {
      "Condition": "CreateWebServer",
      "Value": {
        "Fn::GetAtt": ["WebServer", "PublicIps"]
      }
    }
  }
}
```

Examples

The following example contains two output items.

- The value of the InstanceId parameter of WebServer.
- The values of the PublicIp and PrivateIp parameters of WebServer.

```
"Outputs": {
  "InstanceId": {
    "Value" : {"Fn::GetAtt": ["WebServer", "InstanceId"]}
  },
  "PublicIp & PrivateIp": {
    "Value" : [
      {"Fn::GetAtt": ["WebServer", "PublicIp"]},
      {"Fn::GetAtt": ["WebServer", "PrivateIp"]}
    ]
  }
}
```

5.1.4.5. Functions

Resource Orchestration Service (ROS) provides several built-in functions to help you manage stacks. You can use built-in functions to define Resources and Outputs.

Fn::Base64Encode

The Fn::Base64Encode function is used to return the Base64 representation of the input string.

Declaration

```
"Fn::Base64Encode": "stringToEncode"
```

Parameters

`stringToEncode`: the string decoded from the Base64-encoded string.

Return value

The Base64 representation of the input string.

Examples

```
{"Fn::Base64Encode": "string to encode"}
```

`c3RyaW5nIHRvIGVuY29kZQ==` is returned in this example.

Fn::Base64Decode

The Fn::Base64Decode function is used to return a string decoded from a Base64-encoded string.

Declaration

```
{"Fn::Base64Decode": "stringToEncode"}
```

Parameters

`stringToDecode`: the string decoded from the Base64-encoded string.

Return value

The string decoded from the Base64-encoded string.

Examples

```
{"Fn::Base64Decode": "c3RyaW5nIHRvIGVuY29kZQ=="}
```

`string to encode` is returned in this example.

Fn::Base64

The Fn::Base64 function returns the Base64 representation of the input string.

- Declaration

  `"Fn::Base64": stringToEncode`
- Parameters

  valueToEncode: the string to be encoded in Base64.
- Return value

  The Base64 representation of the input string.
- Examples

  `"Fn::Base64":"string to encode"`

Fn::FindInMap

The Fn::FindInMap function is used to return the values based on keys in a two-level mapping that is declared in the Mappings section.

Declaration

```
"Fn::FindInMap": ["MapName", "TopLevelKey", "SecondLevelKey"]
```

Parameters

- `MapName`: the ID of a mapping declared in the Mappings section that contains keys and values.
- `TopLevelKey`: the top-level key name. The value is a list of key-value pairs.
- `SecondLevelKey`: the second-level key name. The value is a string or a number.

Return value

The value that is assigned to the SecondLevelKey parameter.

Examples

The ImageId property must be specified when you create a WebServer instance. The Mappings section describes the ImageId mappings by region. The Parameters section describes the regions that must be specified by template users. Fn::FindInMap finds the corresponding ImageId mapping in RegionMap based on the region specified by a user, and then finds the corresponding ImageId in the mapping.

- MapName can be set to a custom value, which is `"RegionMap"` in this example.
- TopLevelKey is set to the region where the stack is created, which is `{ "Ref" : "regionParam" }` in this example.
- SecondLevelKey is set to the required architecture, which is `"32"` in this example.

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "regionParam": {
      "Description": "The region where the ECS instance is created",
      "Type": "String",
      "AllowedValues": [
        "hangzhou",
        "beijing"
      ]
    }
  },
  "Mappings": {
    "RegionMap": {
      "hangzhou": {
        "32": "m-25l0rcfjo",
        "64": "m-25l0rcfj1"
      },
      "beijing": {
        "32": "m-25l0rcfj2",
        "64": "m-25l0rcfj3"
      }
    }
  },
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId": {
          "Fn::FindInMap": [
            "RegionMap",
            {"Ref": "regionParam"},
            "32"
          ]
        },
        "InstanceType": "ecs.t1.small",
        "SecurityGroupId": "sg-25zwc****",
        "ZoneId": "cn-beijing-b",
        "Tags": [
          {
            "Key": "key1",
            "Value": "value1"
          },
          {
            "Key": "key2",
            "Value": "value2"
          }
        ]
      }
    }
  }
}
```

Supported functions

- Fn::FindInMap
- Ref

Fn::GetAtt

The Fn::GetAtt function is used to return the value of a property from a resource in a template.

Declaration

```
"Fn::GetAtt": ["resourceID", "attributeName"]
```

Parameters

- `resourceID`: the ID of the resource.
- `attributeName`: the name of the resource property.

Return value

The value of the resource property.

Examples

The ImageId property of MyEcsInstance is returned in this example.

```
{"Fn::GetAtt" : ["MyEcsInstance" , "ImageID"]}
```

Fn::Join

The Fn::Join function is used to append a set of values into a single value that is separated by a specified delimiter.

Declaration

```
{"Fn::Join": ["delimiter", ["string1", "string2", ... ]]}
```

Parameters

- `delimiter`: the value used to divide the string. The delimiter value can be left blank so that all the values are directly combined.
- `[ "string1", "string2", ... ]`: the list of values that are combined into a string.

Return value

The combined string.

Examples

```
{"Fn::Join": [ ",", ["a", "b", "c"]]}
```

`"a,b,c"` is returned in this example.

Supported functions

- Fn::Base64Encode
- Fn::GetAtt
- Fn::Join
- Fn::Select
- Ref

Fn::Select

The Fn::Select function is used to return a single data element from a list of data elements by an index.

Declaration

- The following example assumes that the list of data elements is an array:

```
"Fn::Select": ["index", ["value1", "value2", ... ]]
```

- The following example assumes that the list of data elements is a mapping table:

```
"Fn::Select": ["index", {"key1": "value1", ... }]
```

Parameters

`index`: the index of the object data element. If the list of data elements is an array, the index must be an integer ranging from 0 to N-1, where N indicates the number of elements in the array. If the list of data elements is a mapping table, the index must be a key in the mapping table.

If the corresponding value of the index cannot be found, the system returns an empty string.

Return value

The object data element.

Examples

- The following example assumes that the list of data elements is an array:

```
{"Fn::Select": ["1", ["apples", "grapes", "oranges", "mangoes"]]}
```

`"grapes"` is returned in this example.

- The following example assumes that the list of data elements is a mapping table:

```
{"Fn::Select": ["key1", {"key1": "grapes", "key2": "mangoes"}]}
```

`"grapes"` is returned in this example.

- The following example assumes that the list of data elements is a comma-delimited list:

```
"Parameters": {
  "userParam": {
    "Type": "CommaDelimitedList",
      "Default": "10.0.100.0/24, 10.0.101.0/24, 10.0.102.0/24"
  }
}
"Resources": {
  "resourceID": {
      "Properties": {
          "CidrBlock": {"Fn::Select": ["0", {"Ref": "userParam"}]}
    }
  }
}
```

Supported functions

For the Fn::Select index value, you can use the Ref function.

For the Fn::Select list of data elements, you can use the following functions:

- Fn::Base64Encode
- Fn::FindInMap
- Fn::GetAtt
- Fn::Join
- Fn::Select
- Ref

Ref

The Ref function is used to return the value of a specified parameter or resource.

If the specified parameter is a resource ID, the value of the resource is returned. Otherwise, the system will return the value of the specified parameter.

Declaration

```
"Ref": "logicalName"
```

Parameters

`logicalName`: the logical name of the resource or parameter that you want to reference.

Return value

The value of the resource or parameter.

Examples

The following example uses the Ref function to specify regionParam as the region parameter for RegionMap of WebServer:

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "regionParam": {
      "Description": "The region where the ECS instance is created",
      "Type": "String",
      "AllowedValues": [
        "hangzhou",
        "beijing"
      ]
    }
  },
  "Mappings": {
    "RegionMap": {
      "hangzhou": {
        "32": "m-25l0rcfjo",
        "64": "m-25l0rcfj1"
      },
      "beijing": {
        "32": "m-25l0rcfj2",
        "64": "m-25l0rcfj3"
      }
    }
  },
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId": {
          "Fn::FindInMap": [
            "RegionMap",
            {"Ref": "regionParam"},
            "32"
          ]
        },
        "InstanceType": "ecs.t1.small",
        "SecurityGroupId": "sg-25zwc****",
        "ZoneId": "cn-beijing-b",
        "Tags": [
          {
            "Key": "tiantt",
            "Value": "ros"
          },
          {
            "Key": "tiantt1",
            "Value": "ros1"
          }
        ]
      }
    }
  }
}
```

Supported function

When you use Ref function, you cannot use other functions in it at the same time. You must specify a string value for the resource logical ID.

Fn::GetAZs

The Fn::GetAZs function is used to return a list of zones for a specified region.

Declaration

```
"Fn::GetAZs": "region"
```

Parameters

`region`: the ID of the region.

Return value

The list of zones for the specified region.

Examples

The following example demonstrates how to create an ECS instance in the first zone of a specified region:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Resources" : {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId": "centos7u2_64_40G_cloudinit_2016****.raw",
        "InstanceType": "ecs.n1.tiny",
        "SecurityGroupId": "sg-2zedcm7ep5quses0****",
        "Password": "Ros1****",
        "AllocatePublicIP": true,
        "InternetChargeType": "PayByTraffic",
        "InternetMaxBandwidthIn": 100,
        "InternetMaxBandwidthOut": 100,
        "SystemDiskCategory": "cloud_efficiency",
        "IoOptimized": "optimized",
        "ZoneId": {"Fn::Select": ["0", {"Fn::GetAZs": {"Ref": "ALIYUN::Region"}}]}
      }
    }
  },
  "Outputs": {
    "InstanceId": {
         "Value": {"Fn::GetAtt": ["WebServer","InstanceId"]}
    },
    "PublicIp": {
         "Value": {"Fn::GetAtt": ["WebServer","PublicIp"]}
    }
  }
}
```

Supported functions

- Fn::Base64Encode
- Fn::FindInMap
- Fn::GetAtt
- Fn::Join
- Fn::Select
- Ref

Fn::Replace

The Fn::Replace function is used to replace a specified substring contained in a string with a new substring.

Declaration

```
{"Fn::Replace": [{"object_key": "object_value"}, "object_string"]}
```

Parameters

- `object_key` : the substring to be replaced.
- `object_value` : the new substring to replace the previous substring.
- `object_string` : the string whose `object_key` is replaced.

Return value

The string after replacement.

Examples

The following example demonstrates how to replace "print" with "echo" in the specified script:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Resources" : {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId" : "centos_7_2_64_40G_base_2017****.vhd",
        "InstanceType": "ecs.n1.medium",
        "SecurityGroupId": "sg-94q49****",
        "Password": "MytestPassword****",
        "IoOptimized": "optimized",
        "VSwitchId": "vsw-94vdv8****",
        "VpcId": "vpc-949uz****",
        "SystemDiskCategory": "cloud_ssd",
        "UserData": {"Fn::Replace": [{"print": "echo"},
         {"Fn::Join": ["", [
             "#! /bin/sh\n",
             "mkdir ~/test_ros\n",
             "print hello > ~/1.txt\n"
         ]]}]}
      }
    }
  },
  "Outputs": {
    "InstanceId": {
         "Value" : {"Fn::GetAtt": ["WebServer","InstanceId"]}
    },
    "PublicIp": {
         "Value" : {"Fn::GetAtt": ["WebServer","PublicIp"]}
    }
  }
}
```

Supported functions

- Fn::Base64Encode
- Fn::GetAtt
- Fn::Join
- Fn::Select

- Ref

Fn::Split

The Fn::Split function is used to split a string into a list of values separated by a specified delimiter and return the list.

Declaration

```
"Fn::Split": ["delim",  "original_string"]
```

Parameters

- `delim`: the specified delimiter, which can be commas (,), semicolons (;), line breaks (\n), and indents (\t).
- `original_string`: the string to be split.

Return value

A list of string values.

Examples

- The following example assumes that the list of data elements is an array:

  ```
  {"Fn::Split": [";", "foo; bar; achoo"]}
  ```

  `["foo", " bar", "achoo "]` is returned in this example.

- The following example uses Fn::Split to split InstanceIds:

  ```
  {
    "Parameters": {
      "InstanceIds": {
        "Type": "String",
        "Default": "instane1_id,instance2_id,instance2_id"
      }
    },
    "Resources": {
      "resourceID": {
        "Type": "ALIYUN::SLB::BackendServerAttachment",
        "Properties": {
          "BackendServerList": {
            "Fn::Split": [
              ",",
              {
                "Ref": "InstanceIds"
              }
            ]
          }
        }
      }
    }
  }
  ```

Supported functions

- Fn::Base64Encode
- Fn::FindInMap
- Fn::GetAtt
- Fn::Join
- Fn::Select

- Fn::Replace
- Fn::GetAZs
- Fn::If
- Ref

Fn::Equals

The Fn::Equals function is used to compare whether two values are equal. If the two values are equal, true is returned. If the two values are not equal, false is returned.

Declaration

```
{"Fn::Equals": ["value_1", "value_2"]}
```

Parameters

`value`: the values to be compared.

Return value

true or false.

Examples

The following example uses Fn::Equals to define a condition in the Conditions section:

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "EnvType": {
      "Default": "pre",
      "Type": "String"
    }
  },
  "Conditions": {
    "TestEqualsCond": {
      "Fn::Equals": [
        "prod",
        {"Ref": "EnvType"}
      ]
    }
  }
}
```

Supported functions

- Fn::Or
- Fn::Not
- Fn::Equals
- Fn::FindInMap
- Fn::And
- Ref

Fn::And

The Fn::And function is used to represent the AND operator, and must contain at least two conditions. If all the specified conditions are evaluated as true, true is returned. If any condition is evaluated as false, false is returned.

Declaration

```
{"Fn::And": ["condition", {...}]}
```

Parameters

`condition`: the condition to be evaluated.

Return value

true or false.

Examples

The following example uses Fn::And to define a condition in the Conditions section:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Parameters":{
    "EnvType":{
      "Default":"pre",
      "Type":"String"
    }
  },
  "Conditions": {
    "TestEqualsCond": {"Fn::Equals": ["prod", {"Ref": "EnvType"}]},
    "TestAndCond": {"Fn::And": ["TestEqualsCond", {"Fn::Equals": ["pre", {"Ref": "EnvType"}]}]}
  }
}
```

Supported functions

- Fn::Or
- Fn::Not
- Fn::Equals
- Fn::FindInMap
- Fn::And
- Ref

Fn::Or

The Fn::Or function is used to represent the OR operator, and must contain at least two conditions. If any specified condition is evaluated as true, true is returned. If all the conditions are evaluated as false, false is returned.

Declaration

```
{"Fn::Or": ["condition", {...}]}
```

Parameters

`condition`: the condition to be evaluated.

Return value

true or false.

Examples

The following example uses Fn::Or to define a condition in the Conditions section:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Parameters":{
    "EnvType":{
      "Default":"pre",
      "Type":"String"
    }
  },
  "Conditions": {
    "TestEqualsCond": {"Fn::Equals": ["prod", {"Ref": "EnvType"}]},
    "TestOrCond": {"Fn::And": ["TestEqualsCond", {"Fn::Equals": ["pre", {"Ref": "EnvType"}]}]}
  }
}
```

Supported functions

- Fn::Or
- Fn::Not
- Fn::Equals
- Fn::FindInMap
- Fn::And
- Ref

Fn::Not

The Fn::Not function is used to represent the NOT operator. If a condition is evaluated as false, true is returned. If a condition is evaluated as true, false is returned.

Declaration

```
{"Fn::Not": "condition"}
```

Parameters

`condition`: the condition to be evaluated.

Return value

true or false.

Examples

The following example uses Fn::Not to define a condition in the Conditions section:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Parameters":{
    "EnvType":{
      "Default":"pre",
      "Type":"String"
    }
  },
  "Conditions": {
    "TestNotCond": {"Fn::Not": {"Fn::Equals": ["pre", {"Ref": "EnvType"}]}}
  }
}
```

Supported functions

- Fn::Or
- Fn::Not
- Fn::Equals
- Fn::FindInMap
- Fn::And
- Ref

Fn::If

This function returns one of two possible values. If a specified condition is evaluated as true, one value is returned. If the specified condition is evaluated as false, the other value is returned. The property values of Resources and Outputs in templates support the Fn::If function. You can use the `ALIYUN::NoValue` pseudo parameter as the return value to delete the corresponding property.

Declaration

```
{"Fn::If": ["condition_name", "value_if_true", "value_if_false"]}
```

Parameters

- `condition_name`: the name of the condition in the Conditions section. A condition is referenced by using the condition name.
- `value_if_true`: If the specified condition is evaluated as true, this value is returned.
- `value_if_false`: If the specified condition is evaluated as false, this value is returned.

Examples

The following example demonstrates how to determine whether to create a data disk based on input parameters:

```
{
    "ROSTemplateFormatVersion":"2015-09-01",
    "Parameters":{
        "EnvType":{
            "Default":"pre",
            "Type":"String"
        }
    },
    "Conditions":{
        "CreateDisk":{
            "Fn::Equals":[
                "prod",
                {
                    "Ref":"EnvType"
                }
            ]
        }
    },
    "Resources":{
        "WebServer":{
            "Type":"ALIYUN::ECS::Instance",
            "Properties":{
                "DiskMappings":{
                    "Fn::If":[
                        "CreateDisk",
                        [
                            {
                                "Category":"cloud_efficiency",
```

```
                                "DiskName":"FirstDataDiskName",
                                "Size":40
                            },
                            {
                                "Category":"cloud_ssd",
                                "DiskName":"SecondDataDiskName",
                                "Size":40
                            }
                        ],
                        {
                            "Ref":"ALIYUN::NoValue"
                        }
                    ]
                },
                "VpcId":"vpc-2zew9pxh2yirtzqxd****",
                "SystemDiskCategory":"cloud_efficiency",
                "SecurityGroupId":"sg-2zece6wcqriejf1v****",
                "SystemDiskSize":40,
                "ImageId":"centos_6_8_64_40G_base_2017****.vhd",
                "IoOptimized":"optimized",
                "VSwitchId":"vsw-2zed9txvy7h2srqo6****",
                "InstanceType":"ecs.n1.medium"
            }
        }
    },
    "Outputs":{
        "InstanceId":{
            "Value":{
                "Fn::GetAtt":[
                    "WebServer",
                    "InstanceId"
                ]
            }
        },
        "ZoneId":{
            "Value":{
                "Fn::GetAtt":[
                    "WebServer",
                    "ZoneId"
                ]
            }
        }
    }
}
```

Supported functions

- Fn::Or
- Fn::Not
- Fn::Equals
- Fn::FindInMap
- Fn::And
- Ref

Fn::ListMerge

The Fn::ListMerge function is used to merge multiple lists into one list.

Declaration

```
{"Fn::ListMerge": [["list_1_item_1", "list_1_imte_2", ...], ["list_2_item_1", "list_2_imte_2", ...]]}
```

Parameters

- `["list_1_item_1", "list_1_imte_2", ...]`: the first list to merge.
- `["list_2_item_1", "list_2_imte_2", ...]`: the second list to merge into the first list.

Examples

The following example demonstrates how to attach two ECS instance groups to an SLB instance:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Resources" : {
    "LoadBalancer": {
      "Type": "ALIYUN::SLB::LoadBalancer",
      "Properties": {
        "LoadBalancerName": "ros",
        "AddressType": "internet",
        "InternetChargeType": "paybybandwidth",
      }
    },
    "BackendServer1": {
      "Type": "ALIYUN::ECS::InstanceGroup",
      "Properties": {
        "ImageId" : "m-2ze9uqi7wo61hwep****",
        "InstanceType": "ecs.t1.small",
        "SecurityGroupId": "sg-2ze8yxgempcdsq3****",
        "MaxAmount": 1,
        "MinAmount": 1
      }
    },
    "BackendServer2": {
      "Type": "ALIYUN::ECS::InstanceGroup",
      "Properties": {
        "ImageId" : "m-2ze9uqi7wo61hwep****",
        "InstanceType": "ecs.t1.small",
        "SecurityGroupId": "sg-2ze8yxgempcdsq3iu****",
        "MaxAmount": 1,
        "MinAmount": 1
      }
    },
    "Attachment": {
      "Type": "ALIYUN::SLB::BackendServerAttachment",
      "Properties": {
        "LoadBalancerId": {"Ref": "LoadBalancer"},
        "BackendServerList": { "Fn::ListMerge": [
            {"Fn::GetAtt": ["BackendServer1", "InstanceIds"]},
            {"Fn::GetAtt": ["BackendServer2", "InstanceIds"]}
          ]
        }
      }
    }
  }
}
```

Supported functions

- Fn::Base64Encode
- Fn::GetAtt
- Fn::Join
- Fn::Select
- Ref
- Fn::Join
- Fn::If

Fn::GetJsonValue

The Fn::GetJsonValue function is used to resolve a JSON string and obtain its key value from the first layer.

Declaration

```
{"Fn::GetJsonValue": ["key", "json_string"]}
```

Parameters

- `key`: the key value.
- `json_string`: the specified JSON string to be resolved.

Examples

In the following example, the WebServer instance executes UserData and returns a JSON string, and the WebServer2 instance then obtains the corresponding key value from the string.

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Resources" : {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId" : "m-2ze45uwova5fedlu****",
        "InstanceType": "ecs.n1.medium",
        "SecurityGroupId": "sg-2ze7pxymaix640qr****",
        "Password": "Wenqiao****",
        "IoOptimized": "optimized",
        "VSwitchId": "vsw-2zei67xd9nhcqxze****",
        "VpcId": "vpc-2zevx9ios1rszqv0a****",
        "SystemDiskCategory": "cloud_ssd",
        "UserData": {"Fn::Join": ["", [
             "#! /bin/sh\n",
             "mkdir ~/test_ros\n",
             "print hello > ~/1.txt\n",
             "Fn::GetAtt": ["WaitConHandle", "CurlCli"],
             "\n",
             "Fn::GetAtt": ["WaitConHandle", "CurlCli"],
             " -d '{\\"id\\" : \\"1\\", \\"data\\": [\\"1111\\", \\"2222\\"]}'\n"
         ]]},
         "PrivateIpAddress": "192.168.XX.XX",
         "HostName": "userdata-1
      }
    },
    "WaitConHandle": {
        "Type": "ALIYUN::ROS::WaitConditionHandle"
    },
    "WaitCondition": {
        "Type": "ALIYUN::ROS::WaitCondition",
        "Properties": {
```

```
          "Handle": {"Ref": "WaitConHandle"},
          "Timeout": 900
        }
    },
    "WebServer2": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId" : "m-2ze45uwova5fedlu****",
        "InstanceType": "ecs.n1.medium",
        "SecurityGroupId": "sg-2ze7pxymaix640qr****",
        "Password": "Wenqiao****",
        "IoOptimized": "optimized",
        "VSwitchId": "vsw-2zei67xd9nhcqxzec****",
        "VpcId": "vpc-2zevx9ios1rszqv0a****",
        "SystemDiskCategory": "cloud_ssd",
        "UserData":
          {"Fn::Join": ["", [
             "#! /bin/sh\n",
             "mkdir ~/test_ros\n",
             "echo hello > ~/1.txt\n",
             "server_1_token=",
             {"Fn::GetJsonValue": ["1", { "Fn::GetAtt": ["WaitCondition", "Data"]}]},
             "\n"
         ]]},
         "PrivateIpAddress": "192.168.XX.XX",
         "HostName": "userdata-2"
      }
    },
  },
  "Outputs": {
    "InstanceId": {
         "Value" : {"Fn::GetAtt": ["WebServer","InstanceId"]}
    },
    "PublicIp": {
         "Value" : {"Fn::GetAtt": ["WebServer","PublicIp"]}
    }
  }
}
```

Supported functions

- Fn::Base64Encode
- Fn::GetAtt
- Fn::Join
- Fn::Select
- Ref
- Fn::Join
- Fn::If

Fn::MergeMapToList

The Fn::MergeMapToList function is used to merge multiple mappings into a list of mapping elements.

Declaration

```
{"Fn::MergeMapToList": [{"key_1": ["key_1_item_1", "key_1_item_2", ...]}, {"key_2":["key_2_item_1", "
key_2_item_2", ...]}, ... ]}
```

Parameters

- `{"key_1": ["key_1_item_1", "key_1_item_2", ...]}`: the first mapping to merge. The `" key_1"` value must be a list. `" key_1"` is the key for each mapping in the list of merged mappings. The "key_1" value is `"key_1_item_1"` for the first merged mapping and `"key_1_item_2"` for the second merged mapping. All values follow the same format. The length of the final list of merged mappings is the length of the longest list `"key_x"` from all mappings being merged. If a `"key_y"` list is shorter, the last element of the list is repeated until the list is the longest.
- `{"key_2": ["key_2_item_1", "key_2_item_2", ...]}`: the second mapping to merge into the first mapping. The `" key_2"` value must be a list. `" key_2"` is the key for each mapping in the merged list. The "key_2" value is `"key_2_item_1"` for the first merged mapping and `"key_2_item_2"` for the second merged mapping.

Examples

- The following example demonstrates how to merge three mappings. The length of the list based on the key values for each mapping is the same.

```
{
   "Fn::MergeMapToList": [
      {"key_1": ["kye_1_item_1", "kye_1_item_2"]},
      {"key_2": ["kye_2_item_1", "kye_2_item_2"]},
      {"key_3": ["kye_3_item_1", "kye_3_item_2"]}
   ]
}
```

The following code shows the merged result:

```
[
   {
      "key_1": "kye_1_item_1",
      "key_2": "kye_2_item_1",
      "key_3": "kye_3_item_1"
    },
    {
      "key_1": "kye_1_item_2",
      "key_2": "kye_2_item_2",
      "key_3": "kye_3_item_2"
    }
]
```

- The length of the list based on the key values for each mapping varies in the following example:

```
{
   "Fn::MergeMapToList": [
      {"key_1": ["kye_1_item_1", "kye_1_item_2"]},
      {"key_2": ["kye_2_item_1", "kye_2_item_2", "key_2_item_3"]},
      {"key_3": ["kye_3_item_1", "kye_3_item_2"]}
   ]
}
```

The following code shows the merged result:

```
[
   {
      "key_1": "kye_1_item_1",
      "key_2": "kye_2_item_1",
      "key_3": "kye_3_item_1"
    },
    {
      "key_1": "kye_1_item_2",
      "key_2": "kye_2_item_2",
      "key_3": "kye_3_item_2"
    },
    {
      "key_1": "kye_1_item_2",
      "key_2": "kye_2_item_3",
      "key_3": "kye_3_item_2"
    }
]
```

- In the following template example, all instances created in WebServer are added to the VServer group of an SLB instance:

```
{
    "ROSTemplateFormatVersion": "2015-09-01",
    "Resources": {
        "WebServer": {
          "Type": "ALIYUN::ECS::InstanceGroupClone",
          "Properties": {
              "SourceInstanceId": "i-xxxxx",
               "Password": "Hello****",
               "MinAmount": 1,
               "MaxAmount": 1
          }
        },
        "CreateVServerGroup": {
            "Type": "ALIYUN::SLB::VServerGroup",
            "Properties": {
                "LoadBalancerId": "lb-****",
                "VServerGroupName": "VServerGroup-****",
                "BackendServers": {
                   "Fn::MergeMapToList": [
                       {"Port": [6666, 9090, 8080]},
                       {"ServerId": {"Fn::GetAtt": ["WebServer", "InstanceIds"]}},
                       {"Weight": [20, 100]}
                   ]
                }
            }
        }
    }
}
```

Supported functions

- Fn::Base64Encode
- Fn::GetAtt
- Fn::Join
- Fn::Select
- Ref
- Fn::Join

- Fn::If
- Fn::ListMerge
- Fn::GetJsonValue

Fn::Avg

The Fn::Avg function is used to return the average value of a set of numbers.

Declaration

```
{"Fn::Avg": [ndigits, [number1, number2, ... ]]}
```

Parameters

- `ndigits`: the number of decimal places to report. This parameter value must be an integer.
- `[ number1, number2, ... ]`: the set of numbers for which the average value will be calculated. Each element in the group must be either a number or a string that can be converted into a number.

Return value

The average value of the set of numbers.

Examples

```
{ "Fn::Avg": [ 1, [1, 2, 6.0] ] }
{ "Fn::Avg": [ 1, ['1', '2', '6.0'] ] }
```

3.0 is returned in this example.

Supported functions

- Fn::GetAtt
- Ref

Fn::SelectMapList

The Fn::SelectMapList function is used to return a list of map elements.

Declaration

```
{"Fn::SelectMapList": ["key2", [{"key1": "value1-1", "key3": "value1-3"}, {"key1": "value2-1", "key2"
: "value2-2"}, {"key1": "value3-1", "key2": "value3-2"}, ...] ] }
```

Parameters

- `key2`: the key to be queried in the map.
- `[{ "key1": "value1-1", "key3": "value1-3" }, ... ]`: the list of maps.

Return value

A list of key values for all maps in the map list.

Examples

```
{
 "Fn::SelectMapList": [
   "key2",
   [
      {"key1": "value1-1", "key3": "value1-3"},
      {"key1": "value2-1", "key2": "value2-2"},
      {"key1": "value3-1", "key2": "value3-2"}
   ]
 ]
}
```

`["value2-2","value3-2"]` is returned in this example.

Fn::Add

The Fn::Add function is used to sum the values of parameters.

Declaration

```
{"Fn::Add": [{"Product": "ROS"}, {"Fn": "Add"}]}
```

Parameters

- The parameters must be arranged as a list.
- The parameters in the list can be of the Number, List, or Dictionary type. All the parameters must be of the same type. The list must contain at least two parameters.

Return value

If the parameter values are numbers, sum the parameter values. If the parameter values are lists, concatenate the values. If the parameter values are dictionaries, merge the values. If the two parameters have the same key, overwrite the former parameter value with the latter.

Examples

```
{
 "Fn::Add": [
   {"Product": "ROS"},
   {"Fn": "Add"}
 ]
}
```

`{"Fn":"Add","Product":"ROS"}` is returned in this example.

5.1.4.6. Mappings

The Mappings section is a key-value mapping table. When mappings are used in Resources or Outputs definitions, use Fn::FindInMap to find their values by using corresponding keys.

Syntax

A mapping consists of key-value pairs, where both the keys and values can be strings or numbers. Multiple mappings are separated with commas (,). Each mapping name must be unique. Mappings must be pure data and cannot parse functions.

Examples

The following example shows a correct mapping definition:

```
"Mappings": {
    "ValidMapping": {
        "TestKey1": {"TestValu1": "value1"},
        "TestKey2": {"TestValu2": "value2"},
        1234567890: {"TestValu3": "value3"},
        "TestKey4": {"TestValu4": 1234}
    }
}
```

The following example shows an incorrect mapping definition:

```
"Mappings": {
    "InvalidMapping1": {
        "ValueList": ["foo", "bar"],
          "ValueString": "baz"
    },
    "InvalidMapping2": ["foo", {"bar" : "baz"}],
    "InvalidMapping3": "foobar"
}
```

The following example shows how to use Fn::FindInMap to find the return value:

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "regionParam": {
      "Description": "The region where the ECS instance is created",
      "Type": "String",
      "AllowedValues": [
        "hangzhou",
        "beijing"
      ]
    }
  },
  "Mappings": {
    "RegionMap": {
      "hangzhou": {
        "32": "m-25l0rcfjo",
        "64": "m-25l0rcfj1"
      },
      "beijing": {
        "32": "m-25l0rcfj2",
        "64": "m-25l0rcfj3"
      }
    }
  },
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId": {
          "Fn::FindInMap": [
            "RegionMap",
            {
              "Ref": "regionParam"
            },
            "32"
          ]
        },
        "InstanceType": "ecs.t1.small",
        "SecurityGroupId": "sg-25zwc****",
        "ZoneId": "cn-beijing-b",
        "Tags": [
          {
            "Key": "Department1",
            "Value": "HumanResource"
          },
          {
            "Key": "Department2",
            "Value": "Finance"
          }
        ]
      }
    }
  }
}
```

5.1.4.7. Conditions

There are four conditional operators: Fn::And, Fn::Or, Fn::Not, and Fn::Equals. These operators, along with the parameters that you specify when you create or update a stack, are used to evaluate each condition. You can reference other conditions, parameters, and mappings in your condition. Conditions are used in resource and output definitions to establish dependencies. Use Fn::If or Condition in resource and output definitions to implement conditions.

Syntax

Each condition consists of a condition name and a condition body. The condition name is a string. The condition body starts with Fn::And, Fn::Or, Fn::Not, or Fn::Equals. You can reference other conditions in your condition and separate multiple conditions with commas (,). Each condition name must be unique.

The following functions can be used, but not as the outermost functions:

"Fn::Select", "Fn::Join", "Fn::Split", "Fn::Replace","Fn::Base64Encode", "Fn::Base64Decode", "Fn::MemberListToMap","Fn::If", "Fn::ListMerge", "Fn::GetJsonValue", "Fn::MergeMapToList","Fn::SelectMapList", "Fn::Add", "Fn::Avg", "Fn::Str", "Fn::Calculate","Ref"(parameter references only), and "Fn:: FindInMap".

Examples

The following example shows how to define conditions:

```
"Conditions" : {
       "DevEnv": {"Fn::Equals": ["Dev", {"Ref": "EnvType"}]},
       "UTEnv": {"Fn::Equals": ["UT", {"Ref": "EnvType"}]},
       "PREEnv": {"Fn::Not": {"Fn::Or": ["DevEnv", "UTEnv"]}},
       "ProdEnv": {"Fn::And": [{"Fn::Equals": ["Prod", {"Ref": "EnvType"}]}, "PREEnv"]}
}
```

The following example shows how to use conditions in a resource definition:

In this example, a condition is used to determine whether to create a data disk and an OSS bucket for an ECS instance based on the EnvType value.

```
{
    "ROSTemplateFormatVersion":"2015-09-01",
    "Parameters":{
        "EnvType":{
            "Default":"pre",
            "Type":"String"
        }
    },
    "Conditions":{
        "CreateProdRes":{
            "Fn::Equals":[
                "prod",
                {
                    "Ref":"EnvType"
                }
            ]
        }
    },
    "Resources":{
        "WebServer":{
            "Type":"ALIYUN::ECS::Instance",
            "Properties":{
                "DiskMappings":{
                    "Fn::If":[
                        "CreateProdRes",
                        [
```

```
                            {
                                "Category":"cloud_efficiency",
                                "DiskName":"FirstDataDiskName",
                                "Size":40
                            },
                            {
                                "Category":"cloud_ssd",
                                "DiskName":"SecondDataDiskName",
                                "Size":40
                            }
                        ],
                        {
                            "Ref":"ALIYUN::NoValue"
                        }
                    ]
                },
                "VpcId":"vpc-2zew9pxh2yirtzqxd****",
                "SystemDiskCategory":"cloud_efficiency",
                "SecurityGroupId":"sg-2zece6wcqriejf1v****",
                "SystemDiskSize":40,
                "ImageId":"centos_6_8_64_40G_base_2017****.vhd",
                "IoOptimized":"optimized",
                "VSwitchId":"vsw-2zed9txvy7h2srqo6****",
                "InstanceType":"ecs.n1.medium"
            }
        },
        "OssBucket": {
            "Type": "ALIYUN::OSS::Bucket",
            "Condition": "CreateProdRes",
            "Properties": {
                "AccessControl": "private",
                "BucketName": "myprodbucket"
            }
        }
    },
    "Outputs":{
        "InstanceId":{
            "Value":{
                "Fn::GetAtt":[
                    "WebServer",
                    "InstanceId"
                ]
            }
        },
        "OssDomain":{
            "Condition": "CreateProdRes",
            "Value":{
                "Fn::GetAtt":[
                    "OssBucket",
                    "DomainName"
                ]
            }
        }
    }
}
```

5.1.5. Resource types

5.1.5.1. ECS

5.1.5.1.1. ALIYUN::ECS::AutoSnapshotPolicy

ALIYUN::ECS::AutoSnapshotPolicy is used to create an automatic snapshot policy.

Statement

```
{
  "Type" : "ALIYUN::ECS::AutoSnapshotPolicy",
  "Properties" : {
    "TimePoints" : String,
    "RepeatWeekdays" : String,
    "RetentionDays" : Integer,
    "DiskIds" : String,
    "AutoSnapshotPolicyName" : String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
TimePoints	List	Retained	Yes	The points in time at which automatic snapshots are created. Unit: hours.	Value range:[0, 23], represents 24 time points from 00:00 to 23:00. For example: `1` indicating 01:00. To schedule multiple automatic snapshot creation tasks in a day, you can set the TimePoints parameter as an array. • The maximum number of time points allowed is 24. • Use one format for multiple time points like [0, 1,... 23]. Separate time points with commas (,).

Parameter	Type	Required	Editable	Description	Constraint
RepeatWeekdays	List	Retained	Yes	The days of a week on which automatic snapshots are created.	Value range:[1, 7], `1` indicates Monday. To schedule multiple automatic snapshot creation tasks in a week, you can set the RepeatWeekdays parameter as an array. • You can specify up to 7 days over a one week period. • Use one format for multiple time points like [1, 2,... 7]. Separate the time points with commas (,).
RetentionDays	Integer	Retained	Yes	The number of days for which you want to retain automatic snapshots.	Default value: -1. Valid values: • -1: The automatic snapshots are retained indefinitely. • [1, 65536]: The automatic snapshots are retained for the specified number of days. Default value: -1.

Parameter	Type	Required	Editable	Description	Constraint
DiskIds	List	Retained	Yes	The ID of the destination disk. When you want to apply the automatic snapshot policy to multiple disks, you can set the diskids "d-zzzzzzzzz"]. Separate multiple disk IDs with commas (,).	None
AutoSnapshotPolicyName	String	Yes	True	The name of the automatic snapshot policy.	• The name must be 2 to 128 characters in length • It can contain letters, digits, colons (:), underscores (_), and hyphens (-). • It cannot start with http:// or https://. This parameter is empty by default.

Response parameters

Fn::GetAtt

AutoSnapshotPolicyId: the ID of the automatic snapshot policy.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "AutoSnapshotPolicy": {
      "Type": "ALIYUN::ECS::AutoSnapshotPolicy",
      "Properties": {
        "TimePoints": ["0"],
        "RepeatWeekdays": ["1"],
        "RetentionDays": 10,
        "DiskIds": ["<DiskId>"],
        "AutoSnapshotPolicyName": "MyAutoSnapshotPolicy"
      }
    }
  }
}
```

5.1.5.1.2. ALIYUN::ECS::BandwidthPackage

ALIYUN::ECS::BandwidthPackage is used to create a service plan for a NAT gateway.

Syntax

```
{
  "Type": "ALIYUN::ECS::BandwidthPackage",
  "Properties": {
    "Description": String,
    "NatGatewayId": String,
    "ZoneId": String,
    "BandwidthPackageName": String,
    "Bandwidth": Integer,
    "IpCount": Integer
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
NatGatewayId	String	Yes	No	The ID of the NAT gateway to which you want to bind the service plan.	None
Bandwidth	Integer	Yes	No	The bandwidth.	Valid values: 5 to 5000. Unit: Mbit/s. Default value: 5.

Property	Type	Required	Editable	Description	Constraint
IpCount	Integer	Yes	No	The number of public IP addresses assigned to the NAT gateway.	Valid values: 1 to 5.
Description	String	No	No	The description of the service plan.	The description must be 2 to 256 characters in length.
ZoneId	String	No	No	The ID of the zone where the NAT gateway resides.	None
BandwidthPackageName	String	No	No	The name of the service plan.	The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter.

Response parameters

Fn::GetAtt

- BandwidthPackageId: the ID of the service plan.
- BandwidthPackageIps: all IP addresses included in the service plan.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "BandwidthPackage": {
      "Type": "ALIYUN::ECS::BandwidthPackage",
      "Properties": {
        "BandwidthPackageName": "pkg_2",
        "Description": "my_bandwidth",
        "NatGatewayId": "ngw-h1xox****",
        "IpCount": 2,
        "Bandwidth": 5,
        "ZoneId": "cn-beijing-c"
      }
    }
  },
  "Outputs": {
    "BandwidthPackageId": {
         "Value": {"Fn::GetAttr": ["BandwidthPackage","BandwidthPackageId"]}
    },
    "BandwidthPackageIps": {
         "Value": {"Fn::GetAttr": ["BandwidthPackage","BandwidthPackageIps"]}
    }
  }
}
```

5.1.5.1.3. ALIYUN::ECS::Command

ALIYUN::ECS::Command is used to create a Cloud Assistant command.

Statement

```
{
  "Type": "ALIYUN::ECS::Command",
  "Properties": {
    "Name": String,
    "WorkingDir": String,
    "CommandContent": String,
    "Timeout": Integer,
    "Type": String,
    "Description": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
Name	String	Yes	True	The name of the command, which supports all character sets. The name can be up to 30 characters in length.	None
WorkingDir	String	Yes	True	The working directory on the ECS instance where the command will be run.	None

Parameter	Type	Required	Editable	Description	Constraint
CommandContent	String	Yes	Released	The Base64-encoded content of the command. When you specify request parameters `Type` you must also specify this parameter. The parameter value must be Base64-encoded and cannot exceed 16 KB in size after encoding.	None
Timeout	String	No.	True	The timeout period that is specified for the command to run on ECS instances. Unit: seconds. If the command fails to run within the specified period, the command execution will time out and the process will be forcibly terminated. Default value: 3600.	None
Type	String	No	No	The command type. Valid values: • RunBatScript: Creates a Bat script for a Windows instance. • RunPowerShellScript: Create a PowerShell script to run on a Windows instance. • RunShellScript: Creates a Shell script for Linux-based instances.	None
Description	String	Yes	True	The description of the command, which supports all character sets. The description can be up to 100 characters in length.	None

Response parameters

Fn::GetAtt

CommandId: the ID of the command.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "WorkingDir": {
```

```
      "Type": "String",
      "Description": "The path where command will be executed in the instance."
    },
    "CommandContent": {
      "Type": "String",
      "Description": "The content of command. Content requires base64 encoding. Maximum size support
16KB."
    },
    "Type": {
      "Type": "String",
      "Description": "The type of command."
    },
    "Description": {
      "Type": "String",
      "Description": "The description of command."
    },
    "Timeout": {
      "Type": "Number",
      "Description": "Total timeout when the command is executed in the instance. Input the time unit
as second. Default is 3600s."
    },
    "Name": {
      "Type": "String",
      "Description": "The name of command."
    }
  },
  "Resources": {
    "Command": {
      "Type": "ALIYUN::ECS::Command",
      "Properties": {
        "WorkingDir": {
          "Ref": "WorkingDir"
        },
        "CommandContent": {
          "Ref": "CommandContent"
        },
        "Type": {
          "Ref": "Type"
        },
        "Description": {
          "Ref": "Description"
        },
        "Timeout": {
          "Ref": "Timeout"
        },
        "Name": {
          "Ref": "Name"
        }
      }
    }
  },
  "Outputs": {
    "CommandId": {
      "Description": "The id of command created.",
      "Value": {
        "Fn::GetAtt": [
          "Command",
          "CommandId"
        ]
      }
```

```
      }
    }
  }
}
```

5.1.5.1.4. ALIYUN::ECS::CustomImage

ALIYUN::ECS::CustomImage is used to create a custom image.

Statement

```
{
  "Type": "ALIYUN::ECS::CustomImage",
  "Properties": {
    "Description": String,
    "InstanceId": String,
    "ImageName": String,
    "ImageVersion": String,
    "SnapshotId": String,
    "Tag": List,
    "ResourceGroupId": String,
    "Platform": String,
    "DiskDeviceMapping": List,
    "Architecture": String
  }
}
```

Properties

Parameter	Type	Required or Not	Editable	Description	Constraint
Description	String	Yes	Released	The description of the image.	The description can be up to 256 characters in length. This parameter is empty by default. It cannot start with http:// or https://.
InstanceId	String	Yes	Released	The ID of the ECS instance.	If this parameter is specified, an ECS instance will be used to create the custom image.
ImageName	String	Yes	Released	The name of the image.	The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), and hyphens(-). It must start with a letter but cannot start with http:// or https://.
ImageVersion	String	Yes	Released	The image version.	The image version must be 1 to 40 characters in length.

Parameter	Type	Required or Not	Editable	Description	Constraint
SnapshotId	String	Yes	Released	The ID of the snapshot.	• If this parameter is specified, a snapshot will be used to create the custom image. • If both this parameter and the InstanceId parameter are specified, this parameter will be ignored and an instance will be used to create the custom image.
Tags	List	Erased	Released	The tags of the image.	None
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the custom image belongs.	None
Platform	String	Yes	Released	If you specify a data disk snapshot to be used to create the system disk of the custom image, you must use the Platform parameter to determine the release version of the operating system for the system disk.	None
DiskDeviceMapping	List	Erased	Released	The mappings between images and snapshots.	None
Architecture	String	Yes	Released	If you specify a data disk snapshot to be used to create the system disk of the custom image, you must use the Architecture parameter to determine the architecture of the system disk. Default value: x86_64.	Valid values: • i386 • x86_64

Tag syntax

```
"Tag": [
  {
    "Key": String,
    "Value": String
  }
]
```

Tag properties

Parameter	Type	Required or Not	Editable	Description	Constraint
Key	String	Yes	Released	The tag key of the image.	The tag key cannot be a null string. The key can be up to 64 characters in length. It cannot start with aliyun or acs: and cannot contain http:// or https://.
Value	String	Yes	Released	The tag value of the image.	The tag value can be an empty string. The value can be up to 128 characters in length. It cannot start with aliyun or acs: and cannot contain http:// or https://.

DiskDeviceMapping

```
"DiskDeviceMapping": [
  {
    "Device": String,
    "SnapshotId": String,
    "Size": Integer,
    "DiskType": String
  }
]
```

DiskDeviceMapping properties

Parameter	Type	Required or Not	Editable	Description	Constraint
Device	String	Yes	Released	The device name of disk N in the custom image.	The system allocates a device name in alphabetical order from /dev/xvda to /dev/xvdz.
SnapshotId	String	Yes	Released	The ID of the snapshot that is used to create the custom image.	None

Parameter	Type	Required or Not	Editable	Description	Constraint
Size	String	Optional	Released	The size of disk N. Unit: GiB.	Valid values: 5 to 2000. • The default value is the size of the snapshot specified by the DiskDeviceMapping.N.SnapshotId parameter. • If the DiskDeviceMapping.N.SnapshotId parameter is not specified, the default disk size is 5 GiB. • The disk size must be greater than or equal to the size of the snapshot specified by the DiskDeviceMapping.N.SnapshotId parameter.
DiskType	String	Yes	Released	The type of disk N in the custom image. You can specify this parameter to create the system disk of the custom image from a data disk snapshot. If you do not specify this parameter, the disk type is determined by the corresponding snapshot.	Valid values: • system: indicates a system disk. • data: indicates a data disk.

Response parameters

Fn::GetAtt

ImageId: the ID of the custom image.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceGroup",
      "Properties": {
        "VpcId": "vpc-2zevx9ios1rszqv0a****",
        "MinAmount": 1,
        "SecurityGroupId": "sg-2ze7pxymaix640qr****",
        "ImageId": {
          "Ref": "CustomImage"
        },
        "IoOptimized": "optimized",
        "SystemDisk_Description": "SystemDisk.Description",
        "SystemDisk_DiskName": "SystemDisk.DiskName",
        "SystemDisk_Category": "cloud_ssd",
        "VSwitchId": "vsw-2zei67xd9nhcqxzec****",
        "Password": "Wenqiao****",
        "InstanceType": "ecs.n1.medium",
        "MaxAmount": 1
      }
    },
    "CustomImage": {
      "Type": "ALIYUN::ECS::CustomImage",
      "Properties": {
        "InstanceId": "i-2zefq1f3ynnrr89q****",
        "SnapshotId": "s-2ze0ibk1pvak4mw6****",
        "ImageName": "image-test-****",
        "ImageVersion": "verison-6-1"
      }
    }
  },
  "Outputs": {
    "CustomImage": {
      "Value": {
        "Fn::GetAtt": [
          "CustomImage",
          "ImageId"
        ]
      }
    },
    "InstanceIds": {
      "Value": {
        "Fn::GetAtt": [
          "WebServer",
          "InstanceIds"
        ]
      }
    }
  }
}
```

5.1.5.1.5. ALIYUN::ECS::DedicatedHost

ALIYUN::ECS::DedicatedHost is used to create a dedicated host.

Statement

```
{
  "Type": "ALIYUN::ECS::DedicatedHost",
  "Properties": {
    "DedicatedHostType": String,
    "DedicatedHostName": String,
    "AutoReleaseTime": String,
    "Description": String,
    "AutoPlacement": String,
    "Tags": List,
    "ActionOnMaintenance": String,
    "NetworkAttributesSlbUdpTimeout": Integer,
    "ChargeType": String,
    "ResourceGroupId": String,
    "ZoneId": String,
    "NetworkAttributesUdpTimeout": Integer,
    "Quantity": Integer
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
DedicatedHostType	String	No	No	The dedicated host type.	None
DedicatedHostName	String	Yes	Released	The name of the dedicated host.	• The name must be 2 to 128 characters in length and can contain letters, digits, colons (:), underscores (_), and hyphens (-). • Must start with an uppercase or lowercase letter, and cannot start with `http://` or `https://` the beginning. • It can contain digits, colons (:), underscores (_), and hyphens (-).

Parameter	Type	Required	Editable	Description	Constraint
AutoReleaseTime	String	Yes	Released	The time scheduled for the dedicated host to be automatically released. If you do not specify the AutoReleaseTime parameter, the dedicated host will not be automatically released. • The minimum release time must be at least 30 minutes after the current time. • The maximum release time must be at most three years from the current time. • If the value of `ss` is not `00`, the start time is automatically rounded down to the nearest minute based on the value of `mm`.	None
Description	String	Yes	Released	The description of the dedicated host.	None
ZoneId	String	Yes	Released	The ID of the zone where the dedicated host resides. This parameter is empty by default. If this parameter is not specified, the system will automatically select a zone.	None
ChargeType	String	Yes	Released	The billing method of the dedicated host.	Valid values: PostPaid and pay-as-you-go.

Parameter	Type	Required	Editable	Description	Constraint
AutoPlacement	String	Yes	Released	Specifies whether to add the dedicated host to the resource pool for automatic deployment. If you do not specify a DedicatedHostId when you create an instance on a DDH, Alibaba Cloud automatically selects a DDH from the resource pool to host the instance.	Valid values: • on • off **Note** If you do not specify this parameter, the dedicated host is added to the automatic deployment resource pool. If you do not want to add the dedicated host to the resource pool for automatic deployment, set the value to off.
Tags	List	Erased	Released	The custom tags of the instance.	A maximum of 20 tags are supported. The format is as follows: `[{"Key": "tagKey", "Value": "tagValue"}, {"Key": "tagKey2", "Value": "tagValue2"}]`.

Parameter	Type	Required	Editable	Description	Constraint
ActionOnMaintenance	String	Yes	Released	The method used to migrate the instances on the DDH when the DDH fails or needs to be repaired online.	Valid valus: • Migrate: specifies that the instances are migrated to another physical server and restarted. • Stop: specifies that all the instances on the DDH are stopped. If the DDH cannot be repaired, the instances are migrated to another physical server and restarted. The default value is "Migrate" for a dedicated host and "Stop" for a local disk.
NetworkAttributesSlbUdpTimeout	String	Optional	Released	The timeout period for a UDP session.	Valid values: 15 to 310. Unit: seconds.
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the dedicated host belongs.	None
NetworkAttributesUdpTimeout	String	Optional	Released	The timeout period for UDP sessions that users can access for cloud services running on the dedicated host.	Valid values: 15 to 310. Unit: seconds.
Quantity	String	Optional	Released	The number of DDHs that you want to create this time.	Valid values: 1 to 100. Default value: 1

Response parameters

Fn::GetAtt

- OrderId: the ID of the order.
- DedicatedHostIds: the list of host IDs.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "AutoRenewPeriod": {
      "Type": "Number",
```

```
      "Description": "The time period of auto renew. When the parameter InstanceChargeType is PrePaid
, it will take effect.It could be 1, 2, 3, 6, 12\. Default value is 1.",
      "AllowedValues": [
        1,
        2,
        3,
        6,
        12
      ],
      "Default": 1
    },
    "Description": {
      "Type": "String",
      "Description": "The description of host."
    },
    "ZoneId": {
      "Type": "String",
      "Description": "The zone to create the host."
    },
    "DedicatedHostName": {
      "Type": "String",
      "Description": "The name of the dedicated host, [2, 128] English or Chinese characters. It must
begin with an uppercase/lowercase letter or a Chinese character, and may contain numbers, '_' or '-'.
It cannot begin with http:// or https://."
    },
    "ChargeType": {
      "Type": "String",
      "Description": "Instance Charge type, allowed value: Prepaid and Postpaid. If specified Prepaid
, please ensure you have sufficient balance in your account. Or instance creation will be failure. De
fault value is Postpaid.",
      "AllowedValues": [
        "PrePaid",
        "PostPaid"
      ],
      "Default": "PostPaid"
    },
    "AutoRenew": {
      "Type": "String",
      "Description": "Whether renew the fee automatically? When the parameter InstanceChargeType is P
rePaid, it will take effect. Range of value:True: automatic renewal.False: no automatic renewal. Defa
ult value is False.",
      "AllowedValues": [
        "True",
        "False"
      ],
      "Default": "False"
    },
    "Period": {
      "Type": "Number",
      "Description": "Prepaid time period. Unit is month, it could be from 1 to 9 or 12, 24, 36, 48,
60\. Default value is 1.",
      "AllowedValues": [
        1,
        2,
        3,
        4,
        5,
        6,
        7,
        8
```

```
        9,
        12,
        24,
        36,
        48,
        60
      ],
      "Default": 1
    },
    "DedicatedHostType": {
      "Type": "String",
      "Description": "The instance type of host."
    },
    "PeriodUnit": {
      "Type": "String",
      "Description": "Unit of prepaid time period, it could be Week/Month. Default value is Month.",
      "AllowedValues": [
        "Week",
        "Month"
      ],
      "Default": "Month"
    },
    "AutoReleaseTime": {
      "Type": "String",
      "Description": "Auto release time for created host, Follow ISO8601 standard using UTC time. for
mat is 'yyyy-MM-ddTHH:mm:ssZ'. Not bigger than 3 years from this day onwards"
    }
  },
  "Resources": {
    "Host": {
      "Type": "ALIYUN::ECS::DedicatedHost",
      "Properties": {
          "Description": {
          "Ref": "Description"
        },
        "ZoneId": {
          "Ref": "ZoneId"
        },
        "DedicatedHostName": {
          "Ref": "DedicatedHostName"
        },
        "ChargeType": {
          "Ref": "ChargeType"
        },
          "DedicatedHostType": {
          "Ref": "DedicatedHostType"
        },
        "PeriodUnit": {
          "Ref": "PeriodUnit"
        },
        "AutoReleaseTime": {
          "Ref": "AutoReleaseTime"
        }
      }
    }
  },
  "Outputs": {
    "OrderId": {
      "Description": "The order id list of created instance.",
```

```
      "Value": {
        "Fn::GetAtt": [
          "Host",
          "OrderId"
        ]
      }
    },
    "DedicatedHostIds": {
      "Description": "The host id list of created hosts",
      "Value": {
        "Fn::GetAtt": [
          "Host",
          "DedicatedHostIds"
        ]
      }
    }
  }
}
```

5.1.5.1.6. ALIYUN::ECS::Disk

ALIYUN::ECS::Disk is used to create an ECS Disk.

Statement

```
{
  "Type": "ALIYUN::ECS::Disk",
  "Properties": {
    "DiskName": String,
    "Description": String,
    "Tags": List,
    "AutoSnapshotPolicyId": String,
    "Encrypted": Boolean,
    "ZoneId": String,
    "ResourceGroupId": String,
    "SnapshotId": String,
    "DiskCategory": String,
    "PerformanceLevel": String,
    "DeleteAutoSnapshot": Boolean,
    "Size": Integer
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None
ZoneId	String	No	No	The ID of the zone where the instance resides.	None

Parameter	Type	Required	Editable	Description	Constraint
DiskName	String	Yes	Released	The name of the disk.	• The name must be 2 to 128 characters in length • And can contain letters, digits, periods, underscores (_), and hyphens (-). • It cannot start with http:// or https://. • The disk name will be displayed in the ECS console.
Description	String	Yes	Released	The description of the disk.	• The description must be 2 to 256 characters in length. • Cannot `http://` or `https://` the beginning. • The disk description will be displayed in the ECS console.
Tags	List	Erased	Released	The custom tags of the instance.	Up to four tags are supported. Example values: `[{"Key": "tagKey", "Value": "tagValue"}, {"Key": "tagKey2", "Value": "tagValue2"}]`.
DiskCategory	String	Yes	Released	The type of the data disk.	Value range • cloud: indicates a basic disk. • cloud_efficiency: indicates an ultra disk. • cloud_ssd: indicates a standard SSD. • cloud_essd: enhanced SSD (ESSD) Default value: cloud.

Parameter	Type	Required	Editable	Description	Constraint
SnapshotId	String	Yes	Released	The ID of the snapshot used to create the data disk.	• If both this parameter and 'Size' are specified, the value of this parameter prevails. • The actual size of the created disk is the size of the specified snapshot. • Snapshots created on or before July 15, 2013 cannot be used to create disks.
PerformanceLevel	String	Yes	Released	Specifies the performance level of an ESSD when you create the ESSD.	Default value: PL1. Valid values: • PL1: A single enhanced SSD delivers up to 50,000 random read/write IOPS. • PL2: A single ESSD delivers up to 100,000 random read/write IOPS. • PL3: maximum random read /write IOPS of 100,000 per disk.
Size	String	Optional	Released	The size of the disk. Unit: GiB. The value of this parameter must be equal to or greater than the capacity of the specified snapshot.	Valid values: • cloud: 5 to 2000 • cloud_efficiency: 20 to 32768 • cloud_ssd: 20 to 32768 • cloud_essd: 20 to 32768
AutoSnapshotPolicyId	String	Yes	Released	The ID of each automatic snapshot policy.	None
Encrypted	Boolean	Erased	Released	Specifies whether to encrypt the disk.	Valid values: • true • false Default value: false.

Parameter	Type	Required	Editable	Description	Constraint
DeleteAutoSnapshot	Boolean	Erased	Released	Specifies whether to delete the automatic snapshots of the disk when the disk is released.	Valid values: • true • false Default value: true.

Tags syntax

```
"Tags" : [
  {
    "Value" : String,
    "Key" : String
  }
]
```

Tags properties

Parameter	Type	Required	Editable
Key	String	No	No
Value	String	Yes	Released

Response parameters

Fn::GetAtt

- DiskId: the ID of the disk.
- Status: The Status of the disk.

Sample request

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Resources" : {
    "DataDisk": {
      "Type": "ALIYUN::ECS::Disk",
      "Properties": {
        "Size": 10,
        "ZoneId": "cn-beijing-a",
        "DiskName": "DataDisk",
        "Description": "ECSDataDisk"
      }
    }
  },
  "Outputs": {
    "DiskId": {
         "Value" : {"Fn::GetAtt": ["DataDisk","DiskId"]}
    },
    "Status": {
         "Value" : {"Fn::GetAtt": ["DataDisk","Status"]}
    }
  }
}
```

5.1.5.1.7. ALIYUN::ECS::DiskAttachment

ALIYUN::ECS::DiskAttachment is used to attach an ECS disk.

Statement

```
{
   "Type" : "ALIYUN::ECS::DiskAttachment",
   "Properties" : {
      "DiskId" : String,
      "InstanceId" : String,
      "Device" : String,
      "DeleteWithInstance" : String
   }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
InstanceId	String	No	No	The ID of the instance.	None
DiskId	String	No	No	The ID of the disk.	The disk and the ECS instance must belong to the same zone.

Parameter	Type	Required	Editable	Description	Constraint
Device	String	Yes	Released	The name of the disk.	If you do not set this parameter, the system will automatically allocate a device name in alphabetical order from /dev/xvdb to /dev/xvdz.
DeleteWithInstance	Boolean	Erased	Released	Specifies whether the disk is to be released together with the instance.	Valid values: • true: The disk will be released when the instance is released. • false: The disk will be retained when the instance is released.

Response parameters

Fn::GetAtt

- DiskId: the ID of the disk.
- Status: The Status of the disk.
- The name of the Device: disk.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "DiskAttachment": {
      "Type": "ALIYUN::ECS::DiskAttachment",
      "Properties": {
        "InstanceId": {
          "Ref": "InstanceId"
        },
        "Device": {
          "Ref": "Device"
        },
        "DeleteWithInstance": {
          "Ref": "DeleteWithInstance"
        },
        "DiskId": {
          "Ref": "DiskId"
        }
      }
    }
  },
  "Parameters": {
```

```
    "InstanceId": {
      "Type": "String",
      "Description": "The ID of the instance to attach the disk."
    },
    "Device": {
      "Type": "String",
      "Description": "The device where the volume is exposed on the instance. The device name could be /dev/xvd[a-z]. If this parameter is not specified, the default value will be used."
    },
    "DeleteWithInstance": {
      "Type": "Boolean",
      "Description": "If this parameter is set to true, the disk will be deleted while the instance is deleted. If this parameter is set to false, the disk will be retained after the instance is deleted.",
      "AllowedValues": [
        "True",
        "true",
        "False",
        "false"
      ]
    },
    "DiskId": {
      "Type": "String",
      "Description": "The ID of the disk to be attached."
    }
  },
  "Outputs": {
    "Status": {
      "Description": "The disk status now.",
      "Value": {
        "Fn::GetAtt": [
          "DiskAttachment",
          "Status"
        ]
      }
    },
    "Device": {
      "Description": "The device where the volume is exposed on the ECS instance.",
      "Value": {
        "Fn::GetAtt": [
          "DiskAttachment",
          "Device"
        ]
      }
    },
    "DiskId": {
      "Description": "The ID of the created disk.",
      "Value": {
        "Fn::GetAtt": [
          "DiskAttachment",
          "DiskId"
        ]
      }
    }
  }
}
```

5.1.5.1.8. ALIYUN::ECS::ForwardEntry

ALIYUN::ECS::ForwardEntry is used to configure the DNAT table of a NAT Gateway.

Statement

```
{
  "Type": "ALIYUN::ECS::ForwardEntry",
  "Properties": {
    "ExternalIp": String,
    "ExternalPort": String,
    "ForwardTableId": String,
    "InternalIp": String,
    "IpProtocol": String,
    "InternalPort": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ExternalIp	String	No	No	The public IP address of the NAT gateway.	It must be an IP address that is included in the shared NAT Gateway of the bandwidth plan to which the DNAT table belongs.
ExternalPort	String	No	No	The public port number.	Valid values: 1 to 65535.
ForwardTableId	String	No	No	The ID of the DNAT table.	None
InternalIp	String	No	No	The destination IP address to which the request is forwarded.	This IP address is a private IP address.
IpProtocol	String	No	No	The type of the protocol.	Valid values: TCP, UDP, and Any.
InternalPort	String	No	No	The destination private port.	Valid values: 1 to 65535.

Response parameters

Fn::GetAtt

ForwardEntryId: the ID of each entry in the DNAT table.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ForwardEntry": {
      "Type": "ALIYUN::ECS::ForwardEntry",
      "Properties": {
        "ForwardTableId": "my_forwardtable",
        "ExternalIp": "101.201.XX.XX",
        "ExternalPort": "8080",
        "IpProtocol": "TCP",
        "InternalIp": "10.2.XX.XX",
        "InternalPort": "80"
      }
    }
  },
  "Outputs": {
    "ForwardEntryId": {
      "Value" : {"Fn::GetAttr": ["ForwardEntry","ForwardEntryId"]}
    }
  }
}
```

5.1.5.1.9. ALIYUN::ECS::Instance

ALIYUN::ECS::Instance is used to create an ECS instance.

Statement

```
{
  "Type": "ALIYUN::ECS::Instance",
  "Properties": {
    "RamRoleName": String,
    "IoOptimized": String,
    "PrivateIpAddress": String,
    "KeyPairName": String,
    "SystemDiskDiskName": String,
    "Description": String,
    "Tags": List,
    "HostName": String,
    "ImageId": String,
    "ResourceGroupId": String,
    "VSwitchId": String,
    "Password": String,
    "InstanceType": String,
    "SystemDiskCategory": String,
    "UserData": String,
    "SystemDiskSize": Number,
    "ZoneId": String,
    "VpcId": String,
    "InstanceName": String,
    "InternetMaxBandwidthIn": Integer,
    "DeletionProtection": Boolean,
    "DeploymentSetId": String,
    "SecurityGroupId": String,
    "HpcClusterId": String,
    "SystemDiskDescription": String,
    "DiskMappings": List
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None

Parameter	Type	Required	Editable	Description	Constraint
ImageId	String	No	Yes	The ID of the image used to start the ECS instance. You can use a public image, a custom image, or an Alibaba Cloud Marketplace image.	When editing a template, you can specify the image type and version or only the image type. ROS automatically selects an appropriate public image ID. You can use the wildcard character (*) to represent part of an image ID. Take all Ubuntu public images provided by Alibaba Cloud as an example. You can use one of the following methods to specify the public image ID for the ECS instance: • If you enter ubuntu, the system matches it with the following ID: ubuntu16_0402_64_20G_alibase_20170818.vhd • If you enter ubuntu_14, the system matches it with the following ID: ubuntu_14_0405_64_20G_alibase_20170824.vhd • If you enter ubuntu*14*32, the system matches it with the following ID: ubuntu_14_0405_32_40G_alibase_20170711.vhd • If you enter ubuntu_16_0402_32, the system matches it with the following ID: ubuntu_16_0402_32_40G_alibase_20170711.vhd
InstanceType	String	No	No	The type of the ECS instance.	None
SecurityGroupId	String	No	No	The ID of the security group to which the created instance will belong.	None
Description	String	Yes	Released	The description of created instances.	The description can be up to 256 characters in length.

Parameter	Type	Required	Editable	Description	Constraint
InstanceName	String	Yes	Released	The name of a created instance.	The name can be up to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-).
Password	String	Yes	Released	The password used to log on to the ECS instance.	The characters in length is 8 to 30. And must contain at least one of the following character types: uppercase letters, lowercase letters, digits, and special character. Special characters include ( ) ` ~ ! @ # $ % ^ & * - + = | { } [ ] : ; ' < > , . ? / - / - If you specify the Password parameter in the API request, use HTTPS to secure the API and protect your password.
HostName	String	Yes	Released	The hostname of the instance.	The password must be at least 2 characters in length. It cannot And hyphens (-) cannot start or end the hostname and cannot be used consecutively. On Windows, the hostname can be up to 15 characters in length and can contain letters, digits, and hyphens (-). It cannot contain periods (.) and cannot be composed of only digits. On other OSs such as Linux, the hostname can contain a maximum of 30 characters, including periods (.), each segment can contain uppercase or lowercase letters, digits, and hyphens (-).
PrivateIpAddress	String	Yes	Released	The private IP address of an ECS instance in a VPC. The specified IP address must not be used by other instances in the VPC.	None
InternetMaxBandwidthIn	String	Optional	Released	The maximum inbound bandwidth from the Internet.	Valid values: 1 to 100. Default value: 100. Unit: Mbit/s.

Parameter	Type	Required	Editable	Description	Constraint
IoOptimized	String	Yes	Released	Specifies whether an I/O optimized instance is created.	Valid values: • none (non-I/O optimized) • optimized Default value: none.
DiskMappings	List	Erased	Released	The data disks to be attached to the instance.	A maximum of 16 disks can be attached to each instance.
SystemDiskCategory	String	Yes	Released	The type of the system disk.	Valid values: • cloud • cloud_efficiency • cloud_ssd • ephemeral_ssd
SystemDiskDescription	String	Yes	Released	The description of the ECS instance system disk.	None
SystemDiskDiskName	String	Yes	Released	The name of the ECS instance system disk.	None
SystemDiskSize	Number	No.	True	The size of the system disk. Unit: GB.	Valid values: 40 to 500. If a custom image is used to create a system disk, make sure that the size of the system disk is greater than that of the custom image.
Tags	List	Erased	Released	The custom tags of the instance.	A maximum of 20 tags are supported. The format is as follows: `[{&ldquo;Key”:”tagKey”,”Value”:”tagValue”}, {&ldquo;Key”:”tagKey2”,”Value”:”tagValue2”}]` .
UserData	String	Yes	Released	The user data that you provide when you create ECS instances.	The user data can be up to 16KB in size. You do not need to use Base64. you must use special characters. `\` Escape.
ZoneId	String	Yes	Released	The ID of the zone where the instance resides.	None
HpcClusterId	String	Yes	Released	The ID of the HPC cluster to which the ECS instance belongs.	None

Parameter	Type	Required	Editable	Description	Constraint
VpcId	String	Yes	Released	The ID of the VPC to which the ECS instance belongs.	None
VSwitchId	String	Yes	Released	The ID of the VSwitch for the ECS instance.	None
KeyPairName	String	Yes	Released	The name of the key pair that is used to connect to created ECS instances.	• For Windows-based instances, this parameter is empty by default. • In the Linux, if this parameter is specified, the Password content is still set to the instance, but the Password logon method is disabled by default. The key pair is used to verify the logon.
RamRoleName	String	Yes	Released	The RAM role name of the instance. You can call the ListRoles operation to query the role name.	None
DeletionProtection	Boolean	Erased	Released	The release protection property of created instances. It specifies whether the instances can be released from the ECS console or through the DeleteInstance operation.	Valid values: • True • False
DeploymentSetId	String	Yes	True	Deployment Set ID.	None

DiskMappings syntax

```
"DiskMappings": [
  {
    "Category": String,
    "DiskName": String,
    "Description": String,
    "Device": String,
    "SnapshotId": String,
    "Size": String
  }
]
```

DiskMappings properties

Parameter	Type	Required	Editable	Description	Constraint
Size	String	No	No	The size of data disk N. Unit: GB.	None
Category	String	Yes	Released	The type of the data disk.	Valid values: • cloud • cloud_efficiency • cloud_ssd • ephemeral_ssd Default value: cloud.
DiskName	String	Yes	Released	The name of data disk N.	The name can be up to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-).
Description	String	Yes	Released	The description of data disk N.	Valid values: 2 to 256. Default value: Null.
Device	String	Yes	Released	The device name of the data disk.	If you do not specify this parameter, the system automatically allocates a device name in alphabetical order from /dev/xvdb to /dev/xvdz.
SnapshotId	String	Yes	Released	The ID of the snapshot used to create the data disk.	None

Tags syntax

```
"Tags": [
  {
    "Value": String,
    "Key": String
  }
]
```

Tags properties

Parameter	Type	Required	Editable	Description	Constraint
Key	String	No	No	None	None

Parameter	Type	Required	Editable	Description	Constraint
Value	String	Yes	Released	None	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- InstanceId: the ID of the instance. An ID is a globally unique identifier (GUID) generated by the system for an instance.
- PrivateIp: The private IP address of the instance in a VPC. This parameter takes effect only when the NetworkType parameter is set to VPC.
- InnerIp: The private IP address of the instance in a Classic network. This parameter takes effect only when the NetworkType parameter is set to Classic.
- PublicIp: the public IP address of the instance in a Classic network. This parameter takes effect only when the NetworkType parameter is set to Classic.
- ZoneId: the zone ID.
- HostName: the hostname of the instance.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId": "m-25l0rc****",
        "InstanceType": "ecs.t1.small",
        "SecurityGroupId": "sg-25zwc****",
        "ZoneId": "cn-beijing-b",
        "Tags": [{
            "Key": "tiantt",
            "Value": "ros"
        },{
            "Key": "tiantt1",
            "Value": "ros1"
        }
        ]
      }
    }
  },
  "Outputs": {
    "InstanceId": {
         "Value": {"get_attr": ["WebServer","InstanceId"]}
    },
    "PublicIp": {
         "Value": {"get_attr": ["WebServer","PublicIp"]}
    }
  }
}
```

5.1.5.1.10. ALIYUN::ECS::InstanceClone

ALIYUN::ECS::InstanceClone is used to clone an ECS instance.

Statement

```
{
  "Type": "ALIYUN::ECS::InstanceClone",
  "Properties": {
    "DeletionProtection": Boolean,
    "DiskMappings": List,
    "LoadBalancerIdToAttach": String,
    "Description": String,
    "BackendServerWeight": Integer,
    "Tags": List,
    "SecurityGroupId": String,
    "RamRoleName": String,
    "ImageId": String,
    "ResourceGroupId": String,
    "SpotPriceLimit": String,
    "InstanceChargeType": String,
    "SourceInstanceId": String,
    "Period": Number,
    "SpotStrategy": String,
    "Password": String,
    "InstanceName": String,
    "InternetMaxBandwidthIn": Integer,
    "ZoneId": String,
    "KeyPairName": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None
SourceInstanceId	String	No	No	The ID of the ECS instance to be cloned.	The clone operation clones the specified instance, including its instance type, image, bandwidth billing method, bandwidth limit, and network type. If the source ECS instance belongs to multiple security groups, the cloned instance is added only to the first of these security groups.
BackendServerWeight	String	Optional	Released	The weight of the ECS instance in the Server Load Balancer instance.	Value range:[0, 100]. Default value: 100.
LoadBalancerIdToAttach	String	Yes	Released	The ID of the SLB instance to which the ECS instance is to be attached.	None
Description	String	Yes	Released	The description of created instances.	The description can be up to 256 characters in length.

Parameter	Type	Required	Editable	Description	Constraint
ImageId	String	Yes	True	The ID of the image used to start created instances. You can use a public image, a custom image, or an Alibaba Cloud Marketplace image.	When editing a template, you can specify the image type and version or only the image type. ROS automatically selects an appropriate public image ID. You can use the wildcard character (*) to represent part of an image ID. Take all Ubuntu public images provided by Alibaba Cloud as an example. You can use one of the following methods to specify the public image ID for the ECS instance: If you enter ubuntu, the system matches it with the following ID: ubuntu16_0402_64_20G_alibase_20170818.vhd If you enter ubuntu_14, the system matches it with the following ID: ubuntu_14_0405_64_20G_alibase_20170824.vhd If you enter ubuntu*14*32, the system matches it with the following ID: ubuntu_14_0405_32_40G_alibase_20170711.vhd If you enter ubuntu_16_0402_32, the system matches it with the following ID: ubuntu_16_0402_32_40G_alibase_20170711.vhd
SecurityGroupId	String	Yes	Released	The ID of the security group to which the created instance will belong.	None
InstanceName	String	Yes	Released	The name of a created instance.	The name can be up to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-).

Parameter	Type	Required	Editable	Description	Constraint
Password	String	Yes	Released	The password used to log on to the ECS instance.	The password must be 8 to 30 characters in length. The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include ( ) ` ~ ! @ # $ % ^ & * - + = | { } [ ] : ; < > , . ? / If you specify the password parameter in the API request, use HTTPS to secure the API and protect your password.
DiskMappings	List	Erased	Released	The disks to be attached to created instances.	A maximum of 16 disks can be attached to each instance.
Tags	List	Erased	Released	The custom tags of the instance.	A maximum of 20 tags can be specified in the [{"Key":"tagKey","Value":"tagValue" },{ "Key":"tagKey2","Value":"tagValue 2"}].
ZoneId	String	Yes	Released	The ID of the zone where the instance resides.	None
InstanceChargeType	String	Yes	Released	The billing method of the new ECS instance.	Valid values: PrePaid and PostPaid. Default value: Postpaid. If you set this parameter to Prepaid, make sure that you have sufficient balance in your account. Otherwise, the instance fails to be created.
Period	Number	Erased	Released	The subscription period of the new ECS instance. This parameter is required when the InstanceChargeType parameter is set to PrePaid. This parameter is ignored when the InstanceChargeType parameter is set to PostPaid.	Valid values: 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 24, and 36. Unit: month.

Parameter	Type	Required	Editable	Description	Constraint
KeyPairName	String	Yes	Released	The name of the key pair that is used to connect to created ECS instances.	For Windows-based instances, this parameter is empty by default. In the Linux, if this parameter is specified, the Password content is still set to the instance, but the Password logon method is disabled by default. The key pair is used to verify the logon.
RamRoleName	String	Yes	Released	The RAM role name of the instance. You can call the ListRoles operation to query the role name.	None
SpotPriceLimit	String	Yes	Released	The maximum hourly price of the instance.	Parameter SpotStrategy this parameter takes effect only when the value is SpotWithPriceLimit.
SpotStrategy	String	Yes	Released	The bidding policy for the pay-as-you-go instance.	This parameter is valid only when the InstanceChargeType parameter is set to PostPaid. Valid values: NoSpot: applies to regular pay-as-you-go instances. SpotWithPriceLimit: applies to preemptible instances with a maximum hourly price. SpotAsPriceGo: applies to pay-as-you-go instances priced at the market price at the time of purchase. Default value: NoSpot.
InternetMaxBandwidthIn	String	Optional	Released	The maximum inbound bandwidth from the Internet. Unit: Mbit/s.	Valid values: 1 to 200.
DeletionProtection	Boolean	Erased	Released	Specifies whether to enable instance release protection in the console or by calling an API operation. (DeleteInstance) Release instances.	Valid values: true and false.

DiskMappings syntax

```
"DiskMappings": [
  {
    "Category": String,
    "DiskName": String,
    "Description": String,
    "Device": String,
    "SnapshotId": String,
    "Size": String
  }
]
```

DiskMappings properties

Parameter	Type	Required	Editable	Description	Constraint
Size	String	No	No	The size of data disk N. Unit: GB.	None
Category	String	Yes	Released	The type of the data disk.	Valid values: cloud, cloud_efficiency, cloud_ssd, and ephemeral_ssd Default value: cloud.
DiskName	String	Yes	Released	Disk name.	The name can be up to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-).
Description	String	Yes	Released	The description of data disk N.	The description must be 2 to 256 characters in length. This parameter is empty by default.
Device	String	Yes	Released	The device name of the data disk.	If you do not specify this parameter, the system automatically allocates a device name in alphabetical order from /dev/xvdb to /dev/xvdz.
SnapshotId	String	Yes	Released	The ID of the snapshot used to create the data disk.	None

Tags syntax

```
"Tags": [
  {
    "Value": String,
    "Key": String
  }
]
```

Tags properties

Parameter	Type	Required	Editable	Description	Constraint
Key	String	No	No	The tag key, which cannot be an empty string. It can be up to 64 characters in length, cannot start with acs: or aliyun, and cannot contain http:// or https://.	None
Value	String	Yes	Released	The tag value, which can be an empty string. It can be up to 128 characters in length and cannot start with acs: or aliyun. It cannot contain http:// or https://.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- InstanceId: the ID of the instance. An ID is a globally unique identifier (GUID) generated by the system for an instance.
- PrivateIp: The private IP address of the instance in a VPC. This parameter takes effect only when the NetworkType parameter is set to VPC.
- InnerIp: The private IP address of the instance in a Classic network. This parameter takes effect only when the NetworkType parameter is set to Classic.
- PublicIp: the public IP address of the instance in a Classic network. This parameter takes effect only when the NetworkType parameter is set to Classic.
- ZoneId: the zone ID.
- HostName: the hostname of the instance.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceClone",
      "Properties": {
        "SourceInstanceId": "i-25zsk****",
        "SecurityGroupId": "sg-25zwc****",
        "ZoneId": "cn-beijing-b",
        "DiskMappings": [
            {"Size": 10, "Category": "cloud"},
            {"Size": 10, "Category": "cloud", "SnapshotId": "s-25wsw****"}
        ]
      }
    }
  },
  "Outputs": {
    "InstanceId": {
         "Value": {"get_attr": ["WebServer","InstanceId"]}
    },
    "PublicIp": {
         "Value": {"get_attr": ["WebServer","PublicIp"]}
    }
  }
}
```

5.1.5.1.11. ALIYUN::ECS::InstanceGroup

ALIYUN::ECS::InstanceGroup is used to create an ECS instance group.

Syntax

```
{
  "Type": "ALIYUN::ECS::InstanceGroup",
  "Properties": {
    "SystemDiskAutoSnapshotPolicyId": String,
    "DedicatedHostId": String,
    "LaunchTemplateName": String,
    "RamRoleName": String,
    "IoOptimized": String,
    "PrivateIpAddress": String,
    "KeyPairName": String,
    "SystemDiskDiskName": String,
    "Description": String,
    "Tags": List,
    "HostName": String,
    "ImageId": String,
    "ResourceGroupId": String,
    "VSwitchId": String,
    "EniMappings": List,
    "Password": String,
    "InstanceType": String,
    "MaxAmount": Integer,
    "AutoReleaseTime": String,
    "SystemDiskCategory": String,
    "UserData": String,
    "LaunchTemplateId": String,
    "LaunchTemplateVersion": String,
    "SystemDiskSize": Number,
    "ZoneId": String,
    "VpcId": String,
    "InternetMaxBandwidthIn": Integer,
    "DeletionProtection": Boolean,
    "DeploymentSetId": String,
    "Ipv6AddressCount": Integer,
    "SecurityGroupId": String,
    "HpcClusterId": String,
    "SystemDiskDescription": String,
    "Ipv6Addresses": List,
    "NetworkType": String,
    "DiskMappings": List,
    "SystemDiskPerformanceLevel": String
  }
}
```

Properties

Attribute	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	No	Yes	The ID of the resource group to which a created instance belongs.	None.

Attribute	Type	Required	Editable	Description	Constraint
HpcClusterId	String	No	Yes	The ID of the HPC cluster to which a created instance belongs.	None.
MaxAmount	Integer	Supported	Yes	The maximum number of ECS instances that can be created at a time.	Valid values: 1 to 100. The MaxAmount parameter must be set to a value greater than or equal to the value of MinAmount.
MinAmount	String	Yes	Yes	The minimum number of ECS instances that can be created at a time.	Valid values: 1 to 100. The MinAmount parameter must be set to a value less than or equal to the value of MaxAmount.
Description	String	No	Yes	The description of created instances.	The description can be up to 256 characters in length.
InstanceType	String	Yes	Yes	The type of the ECS instance.	None.
ImageId	String	Yes	Yes	The ID of the image used to start an ECS instance. You can use a public image, a custom image, or an Alibaba Cloud Marketplace image.	You can specify a partial public image ID instead of providing the complete ID. The following example shows how to use a CNAME record: • Ubuntu is specified and ubuntu_16_0402_64_20G_alibase_20170818.vhd are matched. • If ubuntu1432 is specified, ubuntu_14_0405_32_40G_alibase_20170711.vhd is matched.
SecurityGroupId	String	No	No	The ID of the security group to which created instances belong.	None.

Attribute	Type	Required	Editable	Description	Constraint
InstanceName	String	No	No	The name of an instance.	The names can be up to 128 characters in length It can contain English letters, Chinese characters, digits, underscores (_), periods (.), and hyphens (-). By `name_prefix[begin_number,bits]name_suffix` format to specify different instance name for each ECS instance.
Password	String	No	Yes	The password used to log on to created ECS instances.	• The password must be 8 to 30 characters in length • and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. • The following special characters are supported `:( ) '~ ! @ # $ % ^ & * - + = | { } [ ] : ; ' < > , . ? /` If you specify the Password parameter in the API request, use HTTPS to secure the API and protect your password.
HostName	String	No	No	The hostname of created ECS instances.	The hostname must contain at least two characters in length. The periods and hyphens (-) cannot start or end the hostname and cannot be used together.
AutoReleaseTime	String	No	No	The time scheduled for created ECS instances to be automatically released.	The time format must comply with ISO8601 specifications, for example, `"yyyy-MM-ddTHH:mm:ssZ"`. The maximum release time must be within three years from the current time.

Attribute	Type	Required	Editable	Description	Constraint
PrivateIpAddress	String	No	No	The private IP address of an ECS instance in a VPC.	The specified IP address must not be used by other instances in the VPC.
DiskMappings	List	No	Yes	The data disks to be attached to created instances.	None.
InternetMaxBandwidthIn	Integer	No	No	The maximum inbound bandwidth from the Internet.	Unit: Mbit/s. Valid values: 1 to 100 Default value: 100.
IoOptimized	String	No	No	Specifies whether the created instances are I/O optimized.	Valid values: • none (non-I/O optimized) • optimized Default value: none.
SystemDiskCategory	String	No	Yes	The category of the system disk.	Valid values: cloud: basic disk cloud_efficiency: the ultra disk cloud_ssd: standard SSDs cloud_essd: enhanced SSD ephemeral_ssd: local SSDs
SystemDiskDescription	String	No	Yes	The description of the ECS instance system disk.	None.
SystemDiskDiskName	String	No	Yes	The name of the ECS instance system disk.	None.
SystemDiskSize	Number	No	Yes	The size of the system disk.	Valid values: 40 to 500. If a custom image is used to create a system disk, make sure that the size of the system disk is greater than that of the custom image.

Attribute	Type	Required	Editable	Description	Constraint
Tags	List	No	Yes	The custom tags of a created instance.	A maximum of 20 tags are supported. The format is as follows: `[{"Key":"tagKey","Value":"tagValue"},{"Key":"tagKey2","Value":"tagValue2"}]`.
UserData	String	No	Yes	The user data that you provide when you create ECS instances.	The user data can be up to 16 KB in size. You do not need to use Base64 for transcoding. Special characters need to be escaped.
ZoneId	String	No	No	The zone ID of the disk.	None.
VpcId	String	No	No	The ID of the VPC.	None.
VSwitchId	String	No	No	The ID of the VSwitch for the ECS instance.	None.
KeyPairName	String	No	Yes	The name of the key pair that is used to connect to created ECS instances.	For Windows-based ECS instances, this parameter is ignored. and it is empty by default. For Linux-based ECS instances, the Password parameter still takes effect if this parameter is specified. However, logon by password is disabled, and the KeyPairName value is used.
RamRoleName	String	No	Yes	The name of the instance RAM role.	You can call the ListRoles operation to query the role name.
DedicatedHostId	String	No	No	The ID of the dedicated host.	None.
LaunchTemplateName	String	No	Yes	The name of the launch template for the instance.	None.

Attribute	Type	Required	Editable	Description	Constraint
EniMappings	List	No	Yes	The elastic network interfaces (ENIs) to be attached to created instances.	Only one ENI can be attached to each instance.
LaunchTemplateId	String	No	Yes	The ID of the launch template.	None.
LaunchTemplateVersion	String	No	Yes	The version of the launch template.	If you do not specify a version, the default version is used.
NetworkType	String	No	No	The network type of created ECS instances.	Valid values: • vpc • classic Default value: classic.
DeletionProtection	Boolean	No	No	The release protection attribute of the instance. It specifies whether you can use the ECS console or call the DeleteInstance operation to release the instance.	Valid values: • true • false
DeploymentSetId	String	No	Yes	Deployment Set ID.	None.

DiskMappings syntax

```
"DiskMappings": [
  {
    "Category": String,
    "DiskName": String,
    "Description": String,
    "Device": String,
    "SnapshotId": String,
    "Size": String,
    "Encrypted": String,
    "KMSKeyId": String,
    "PerformanceLevel": String,
    "AutoSnapshotPolicyId": String
  }
]
```

DiskMappings properties

Attribute	Type	Required	Editable	Description	Constraint
Size	String	Yes	No	The size of a data disk.	Unit: GB.
Category	String	No	No	The type of the data disk.	Valid values: • cloud • cloud_efficiency • cloud_ssd • cloud_essd • ephemeral_ssd For I/O optimized instances, the default value is cloud_efficiency. For non-I/O optimized instances, the default value is cloud.
DiskName	String	No	No	The name of data disk N.	The name can be up to 128 characters in length. It can contain English letters, Chinese characters, digits, underscores (_), periods (.), and hyphens (-).
Description	String	No	No	The description of data disk N.	The description must be 2 to 256 characters in length. The description cannot start with `http://` or `https://`.
Device	String	No	No	The device name of the data disk.	The system allocates a device name in alphabetical order from /dev/xvda to /dev/xvdz.
SnapshotId	String	No	No	The ID of the snapshot.	None.
Encrypted	Boolean	No	No	Specifies whether to encrypt the data disk.	Valid values: • true • false Default value: false
KMSKeyId	String	No	No	The ID of the KMS key corresponding to the data disk.	None.

Attribute	Type	Required	Editable	Description	Constraint
AutoSnapshotPolicyId	String	No	Yes	The ID of the automatic snapshot policy.	None.
PerformanceLevel	String	No	No	The performance level of the enhanced SSD used as the data disk.	• (Default): Maximum random read /write IOPS of 50,000 per disk • PL2: A single enhanced SSD delivers up to 100,000 random read/write IOPS. • PL3: A single enhanced SSD delivers up to 1,000,000 random read/write IOPS.

Tags syntax

```
"Tags": [
  {
    "Value": String,
    "Key": String
  }
]
```

Tags properties

Attribute	Type	Required	Editable	Description	Constraint
Key	String	Yes	No	The key of tag N.	It must be 1 to 128 characters in length, and cannot start with `aliyun` and `acs:` beginning, cannot contain `http://` or `https://`.
Value	String	No	No	The value of tag N.	It must be 0 to 128 characters in length and cannot start with `aliyun` and `acs:` beginning, cannot contain `http://` or `https://`.

EniMappings syntax

```
"EniMappings": [
  {
    "SecurityGroupId": String,
    "VSwitchId": String,
    "Description": String,
    "NetworkInterfaceName": String,
    "PrimaryIpAddress": String
  }
]
```

EniMappings properties

Attribute	Type	Required	Editable	Description	Constraint
SecurityGroupId	String	Yes	Yes	The ID of the security group to which an instance belongs.	The security group and the instance must be in the same VPC.
VSwitchId	String	Yes	No	The ID of the VSwitch to which the instance is connected.	None.
Description	String	No	Yes	The description of the ENI.	It can contain 2 to 256 English letters or Chinese character. It cannot start with `http://` or `https://` the beginning.
NetworkInterfaceName	String	No	Yes	The ENI name.	• The name must be 2 to 128 characters in length • Must start with an uppercase or lowercase letter, and cannot start with `http://` or `https://` the beginning. • It must start with a letter and cannot start with http:// or https://.
PrimaryIpAddress	String	No	No	The primary private IP address of ENIN.	The specified IP address must be available within the CIDR block of the VSwitch. If this parameter is not specified, an available IP address in the VSwitch CIDR block will be selected at random.

Return value

Fn::GetAtt

- InstanceIds: the IDs of created instances in the ECS instance group. An ID is a system-generated globally unique identifier (GUID) for an instance.
- PrivateIps: the list of private IP addresses of instances in a VPC. This parameter takes effect only when the NetworkType parameter is set to VPC. For example, a Json-formatted Array: `["172.16.XX.XX", "172.16.XX.XX",&hellip; "172.16.XX.XX"]` the maximum number of IP addresses that can be specified. Separate multiple IP addresses with commas (,).
- InnerIps: the list of private IP addresses of instances in a classic network. This parameter takes effect only when the NetworkType parameter is set to Classic. For example, a Json-formatted Array: `["10.1.XX.XX", "10.1.XX.XX",&hellip; "10.1.XX.XX"]` the maximum number of IP addresses that can be specified. Separate multiple IP addresses with commas (,).
- PublicIps: the list of public IP addresses of instances in a classic network. This parameter takes effect only when the NetworkType parameter is set to Classic. For example, a Json-formatted Array: `["42.1.XX.XX", "42.1.XX.XX",&hellip; "42.1.XX.XX"]` the maximum number of IP addresses that can be specified. Separate multiple IP addresses with commas (,).
- HostNames: the list of hostnames of all instances.
- OrderId: the list of order IDs of all instances.
- ZoneIds: the IDs of the zones where created instances reside.
- RelatedOrderIds: the list of related order IDs of created ECS instances.

Examples

```
{
  "ROSTemplateFormatVersion":"2015-09-01",
  "Resources":{
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceGroup",
      "Properties": {
        "ImageId":"m-2510r****",
        "InstanceType": "ecs.t1.small",
        "SecurityGroupId": "sg-25zwc****",
        "ZoneId": "cn-beijing-b",
        "MaxAmount":1,
        "MinAmount":1,
        "Tags": [{
            "Key": "tiantt",
            "Value": "ros"
        },{
            "Key": "tiantt1",
            "Value": "ros1"
        }
        ]
      }
    }
  },
  "Outputs": {
    "InstanceIds": {
         "Value":{"get_attr": ["WebServer","InstanceIds"]}
    },
    "PublicIps": {
         "Value":{"get_attr": ["WebServer","PublicIps"]}
    }
  }
}
```

5.1.5.1.12. ALIYUN::ECS::InstanceGroupClone

ALIYUN::ECS::InstanceGroupClone is used to clone an ECS instance group.

Statement

```
{
  "Type": "ALIYUN::ECS::InstanceGroupClone",
  "Properties": {
    "BackendServerWeight": Integer,
    "DiskMappings": List,
    "LaunchTemplateName": String,
    "SpotPriceLimit": String,
    "ResourceGroupId": String,
    "KeyPairName": String,
    "SystemDiskDiskName": String,
    "PeriodUnit": String,
    "Description": String,
    "Tags": List,
    "ImageId": String,
    "SpotStrategy": String,
    "SourceInstanceId": String,
    "EniMappings": List,
    "Password": String,
    "MaxAmount": Integer,
    "AutoReleaseTime": String,
    "SystemDiskCategory": String,
    "LoadBalancerIdToAttach": String,
    "LaunchTemplateId": String,
    "LaunchTemplateVersion": String,
    "ZoneId": String,
    "InstanceName": String,
    "InternetMaxBandwidthIn": Integer,
    "DeletionProtection": Boolean,
    "DeploymentSetId": String,
    "SecurityGroupId": String,
    "RamRoleName": String,
    "HpcClusterId": String,
    "SystemDiskDescription": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the enterprise resource group to which a created instance belongs.	None
HpcClusterId	String	Yes	True	The ID of the E-HPC cluster to which a created instance belongs.	None

Parameter	Type	Required	Editable	Description	Constraint
SourceInstanceId	String	No	No	The ID of the ECS instance to be cloned.	The clone operation clones the specified instance, including its instance type, image, bandwidth limit, and network type. If the source ECS instance belongs to multiple security groups, the cloned instance is added only to the first of these security groups.
MaxAmount	Integer	Retained	Yes	The maximum number of ECS instances to be created.	Valid values: 1 to 100. The MaxAmount parameter must be set to a value greater than or equal to the value of the MinAmount parameter.
MinAmount	String	No	Yes	The minimum number of ECS instances to be created.	Valid values: 1 to 100. The MinAmount parameter must be set to a value less than or equal to the value of the MaxAmount parameter.
BackendServerWeight	String	Optional	Released	The weight assigned to the ECS instance in the Server Load Balancer instance.	Valid values: 0 to 100. Default value: 100.
LoadBalancerIdToAttach	String	Yes	Released	The ID of the SLB instance to which the ECS instance is to be attached.	None
Description	String	Yes	Released	The description of created instances.	The description can be up to 256 characters in length.

Parameter	Type	Required	Editable	Description	Constraint
ImageId	String	Yes	True	The ID of the image used to start created instances. You can use a public image, a custom image, or an Alibaba Cloud Marketplace image.	You can specify a partial public image ID instead of providing the complete ID. When editing a template used to deploy an ECS instance, you can specify the image type and version or only the image type. ROS automatically selects an appropriate public image ID. You can use the wildcard character (*) to represent part of an image ID. You can use one of the following methods to specify the public image ID for the ECS instance: • If you set the parameter to ubuntu, ubuntu_16_0402_64_20G_alibase_20170818.vhd. • If this parameter is set to ubuntu_14, ubuntu_14_0405_64_20G_alibase_20170824.vhd is returned. • Specify: ubuntu1432, which will eventually match: ubuntu_14_0405_32_40G_alibase_20170711.vhd • Specify: ubuntu_16_0402_32, which will eventually match: ubuntu_16_0402_32_40G_alibase_20170711.vhd
SecurityGroupId	String	Yes	Released	The ID of the security group to which created instances belong.	None
InstanceName	String	Yes	Released	The name of a created instance.	The name can be up to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-).
Password	String	Yes	Released	The password used to log on to the ECS instance.	The password The password must be 8 to 30 characters in length and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special character. Special characters include ( ) ` ~ ! @ # $ % ^ & * - + = | { } [ ] : ; ' < > , . ? / If the Password parameter is specified, you must use HTTPS to call the operation to ensure that the Password remains confidential.
DiskMappings	List	Erased	Released	The data disks to be attached to the instance.	A maximum of 16 disks can be attached to each instance.

Parameter	Type	Required	Editable	Description	Constraint
Tags	List	Erased	Released	The custom tags of the instance.	You can specify a maximum of 20 tags. The format is as follows: `[{"Key":"tagKey","Value":"tagValue"},{"Key":"tagKey2","Value":"tagValue2"}]`.
ZoneId	String	Yes	Released	The ID of the zone where the instance resides.	None
KeyPairName	String	Yes	Released	The name of the key pair that is used to connect to the ECS instance. For Windows-based ECS instances, this parameter is ignored. Default value: empty. For Linux-based instances, the Password parameter still takes effect if this parameter is specified. However, logon by Password is disabled, and the KeyPairName value is used.	None
RamRoleName	String	Yes	Released	The RAM role name of the instance.	You can use RAM API ListRoles you can call this operation to query the RAM role name of an instance.
SpotPriceLimit	String	Yes	Released	The maximum hourly price of the instance.	This parameter supports up to three decimal places. Parameter SpotStrategy this parameter takes effect only when the value is SpotWithPriceLimit.
SystemDiskDiskName	String	Yes	True	The name of the system disk of created instances.	The name must be 2 to 128 characters in length Must start with an uppercase or lowercase letter, and cannot start with `http://` or `https://` and can contain digits, colons (:), underscores (_), and hyphens (-).

Parameter	Type	Required	Editable	Description	Constraint
PeriodUnit	String	Yes	True	The billing cycle for created ECS instances.	Valid values: • Week. 1, 2, 3, and 4} when the value of the PeriodUnit parameter is Week. AutoRenewPeriod values are 1, 2, "3"}. • Month 1, 2, 3, 4, 5, 6, and 7 when the PeriodUnit parameter is set to Month "8", "9", "12", "24", "36", "48", "60"},AutoRenewPeriod can be {"1", "2", "3", "6", "12"}. Default value: Month.
EniMappings	List	No.	True	The elastic network interfaces (ENIs) to be attached to a created instance.	Only a single ENI can be attached to each instance.

Parameter	Type	Required	Editable	Description	Constraint
AutoReleaseTime	String	Yes	Released	The time scheduled for a created ECS instance to be automatically released. Specify the time in the ISO 8601 standard in the YYYY-MM-DDThh:mmZ format. in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC. • If the value of seconds (ss) is a value other than 00, the start time is automatically rounded down to the nearest minute based on the value of mm. • The minimum release time must be at least 30 minutes later than the current time. • The maximum release time must be at most three years from the current time. If you do not specify the AutoReleaseTime it indicates that the auto release feature is disabled and the ECS instance will not be automatically released.	None
SystemDiskCategory	String	Yes	True	The type of the system disk.	Valid values: • cloud: basic disk • cloud_efficiency: indicates an ultra disk. • cloud_ssd: indicates a standard SSD. • ephemeral_ssd: local SSD. • cloud_essd: indicates an enhanced SSD (ESSD). ESSDs are still in public preview and only available in some regions. For phased-out instance types that are not I/O optimized, the default value is cloud. For other instances, the default value is cloud_efficiency.

Parameter	Type	Required	Editable	Description	Constraint
LaunchTemplateName	String	Yes	True	The name of the launch template.	None
LaunchTemplateVersion	String	Yes	True	The version of the launch template. If you do not specify this parameter, the default version is used.	None
InternetMaxBandwidthIn	String	Optional	Released	The maximum inbound bandwidth from the Internet. Unit: Mbit/s.	Valid values: 1 to 200. Default value: 200.
LaunchTemplateId	String	Yes	True	The ID of the launch template.	None
SystemDiskDescription	String	Yes	True	The description of the system disk.	The description must be 2 to 256 characters in length and cannot start with http:// or https://. This parameter is empty by default.
DeletionProtection	Boolean	Erased	Released	The release protection attribute of the instance. Specifies whether the ECS console or API DeleteInstance) to release the instance.	Valid values: • true • false
DeploymentSetId	String	Yes	True	Deployment Set ID.	None

DiskMappings syntax

```
"DiskMappings": [
  {
    "Category": String,
    "DiskName": String,
    "Description": String,
    "Encrypted": String,
    "KMSKeyId": String,
    "Device": String,
    "SnapshotId": String,
    "Size": String
  }
]
```

DiskMappings properties

Parameter	Type	Required	Editable	Description	Constraint

Parameter	Type	Required	Editable	Description	Constraint
Encrypted	String	Yes	Released	Specifies whether to encrypt the data disk.	Valid values: • true • false Default value :false
KMSKeyId	String	Yes	Released	The KMS key ID for data disk N.	None
Size	String	No	No	The size of data disk N. Unit: GB.	None
Category	String	Yes	Released	The type of the data disk.	Valid values: • cloud • cloud_efficiency • cloud_ssd • ephemeral_ssdDefault
DiskName	String	Yes	Released	The name of data disk N.	The name can be up to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-).
Description	String	Yes	Released	The description of data disk N.	Valid values: 2 to 256. Default value: Null.
Device	String	Yes	Released	The device name of the data disk attached to an ECS instance.	The system allocates a device name in alphabetical order from /dev/xvda to /dev/xvdz.
SnapshotId	String	Yes	Released	Create a data disk by using a snapshot.	None

EniMappings syntax

```
"EniMappings": [
  {
    "SecurityGroupId": String,
    "VSwitchId": String,
    "Description": String,
    "NetworkInterfaceName": String,
    "PrimaryIpAddress": String
  }
]
```

EniMappings properties

Parameter	Type	Required	Editable	Description	Constraint

Parameter	Type	Required	Editable	Description	Constraint
SecurityGroupId	String	No	Yes	The ID of the security group to which the ENI belongs.	None
VSwitchId	String	No	No	The ID of the VSwitch to which the ENI belongs.	None
Description	String	Yes	True	The description of the ENI.	It can contain 2 to 256 English letters or Chinese character. It cannot start with `http://` and `https://` the beginning.
NetworkInterfaceName	String	Yes	True	The name of the ENI.	None
PrimaryIpAddress	String	Yes	Released	The primary IP address of the ENI.	None

Tags syntax

```
"Tags": [
  {
    "Value": String,
    "Key": String
  }
]
```

Tags properties

Parameter	Type	Required	Editable	Description	Constraint
Key	String	No	No	None	None
Value	String	Yes	Released	None	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- InstanceIds: the IDs of instances in the ECS instance group. An ID is a globally unique identifier (GUID) generated by the system for an instance.
- PrivateIps: the private IP addresses of VPC-type instances. This parameter is valid only when the NetworkType parameter is set to VPC. The parameter value is a JSON-formatted array, containing up to 100 IP addresses separated by commas (,). Example: ["172.16.XX.XX", "172.16.XX.XX", ... "172.16.XX.XX"].
- InnerIps: the private IP addresses of instances in the classic network. This parameter is valid only when the

NetworkType parameter is set to Classic. The parameter value is a JSON-formatted array, containing up to 100 IP addresses separated by commas (,). Example: ["10.1.XX.XX", "10.1.XX.XX", ... "10.1.XX.XX"].

- PublicIps: the public IP addresses of instances in the classic network. This parameter is applicable only when the NetworkType parameter is set is Classic. The parameter value is a JSON-formatted array, containing up to 100 IP addresses separated by commas (,). Example: ["42.1.XX.XX", "42.1.XX.XX", ... "42.1.XX.XX"].
- HostNames: the hostnames of all instances. The parameter value is a JSON-formatted array. Example: ["host1", "host2", ... "host3"].
- OrderId: the order IDs of all instances.
- ZoneIds: the IDs of the zones where created instances reside.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceGroupClone",
      "Properties": {
        "SourceInstanceId": "i-25zsk****",
        "ImageId": "m-25l0r****",
        "SecurityGroupId": "sg-25zwc****",
        "ZoneId": "cn-beijing-b",
        "MaxAmount": 1,
        "MinAmount": 1
      }
    }
  },
  "Outputs": {
    "InstanceIds": {
      "Value": {"get_attr": ["WebServer","InstanceIds"]}
    },
    "PublicIps": {
      "Value": {"get_attr": ["WebServer","PublicIps"]}
    }
  }
}
```

5.1.5.1.13. ALIYUN::ECS::Invocation

ALIYUN::ECS::Invocation is used to invoke a Cloud Assistant command for one or more ECS instances.

Statement

```
{
  "Type": "ALIYUN::ECS::Invocation",
  "Properties": {
    "Timed": Boolean,
    "Frequency": String,
    "CommandId": String,
    "InstanceIds": List
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
Timed	Boolean	Erased	Released	Specifies whether to invoke the command on a periodic basis. Default value: false.	None
Frequency	String	Yes	Released	The frequency at which the command is invoked.	None
CommandId	String	No	No	The ID of the script.	None
InstanceIds	List	Yes	No	The IDs of the instances for which you want to invoke the command. A maximum of 20 instance IDs can be specified.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

The execution ID of the InvokeId: command.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "Timed": {
      "Type": "Boolean",
      "Description": "Whether it is timed execution. Default is False.",
      "AllowedValues": [
        "True",
        "true",
        "False",
        "false"
      ]
    },
    "Frequency": {
      "Type": "String",
      "Description": "The frequency of timing execution (the shortest frequency is performed every 1 minute). It is mandatory when Timing is True.The value rule follows the rules of the cron expression. "
    },
    "CommandId": {
      "Type": "String",
      "Description": "The id of command."
    },
```

```
    "InstanceIds": {
      "Type": "CommaDelimitedList",
      "Description": "The instance id list. Select up to 20 instances at a time.Instances selected network type must be VPC network, status must be running"
    }
  },
  "Resources": {
    "Invocation": {
      "Type": "ALIYUN::ECS::Invocation",
      "Properties": {
        "Timed": {
          "Ref": "Timed"
        },
        "Frequency": {
          "Ref": "Frequency"
        },
        "CommandId": {
          "Ref": "CommandId"
        },
        "InstanceIds": {
          "Fn::Split": [
            ",",
            {
              "Ref": "InstanceIds"
            },
            {
              "Ref": "InstanceIds"
            }
          ]
        }
      }
    }
  },
  "Outputs": {
    "InvokeId": {
      "Description": "The id of command execution.",
      "Value": {
        "Fn::GetAtt": [
          "Invocation",
          "InvokeId"
        ]
      }
    }
  }
}
```

5.1.5.1.14. ALIYUN::ECS::JoinSecurityGroup

ALIYUN::ECS::JoinSecurityGroup is used to add one or more ECS instances to a specified security group.

Syntax

```
{
  "Type": "ALIYUN::ECS::JoinSecurityGroup",
  "Properties": {
    "InstanceId": String,
    "InstanceIdList": List,
    "SecurityGroupId": String,
    "NetworkInterfaceList": List
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
SecurityGroupId	String	Yes	No	The ID of the security group.	None
InstanceId	String	No	No	The ID of the ECS instance to be added to the security group.	None
InstanceIdList	List	No	Yes	The IDs of the ECS instances to be added to the security group.	None
NetworkInterfaceList	List	No	Yes	The IDs of the elastic network interfaces (ENIs).	None

Response parameters

Fn::GetAtt

None

Examples

`JSON` format

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SG": {
      "Type": "ALIYUN::ECS::JoinSecurityGroup",
      "Properties": {
        "SecurityGroupId": "sg-m5eagh7rzys2z8sa****",
        "InstanceIdList": [
          "i-m5e505h9bgsio0wy****",
          "i-m5e505hio0wyjc6r****"
        ]
      }
    }
  }
}
```

`YAML` format

```
ROSTemplateFormatVersion: '2015-09-01'
Resources:
  SG:
    Type: ALIYUN::ECS::JoinSecurityGroup
    Properties:
      SecurityGroupId: sg-m5eagh7rzys2z8sa****
      InstanceIdList:
      - i-m5e505h9bgsio0wy****
      - i-m5e505hio0wyjc6r****
```

5.1.5.1.15. ALIYUN::ECS::LaunchTemplate

ALIYUN::ECS::LaunchTemplate is used to create an ECS instance launch template.

Syntax

```
{
  "Type": "ALIYUN::ECS::LaunchTemplate",
  "Properties": {
    "LaunchTemplateName": String,
    "VersionDescription": String,
    "ImageId": String,
    "InstanceType": String,
    "SecurityGroupId": String,
    "NetworkType": String,
    "VSwitchId": String,
    "InstanceName": String,
    "Description": String,
    "InternetMaxBandwidthIn": Integer,
    "InternetMaxBandwidthOut": Integer,
    "HostName": String,
    "ZoneId": String,
    "SystemDiskCategory": String,
    "SystemDiskSize": Number,
    "SystemDiskDiskName": String,
    "SystemDiskDescription": String,
    "IoOptimized": String,
    "InternetChargeType": String,
    "UserData": String,
    "KeyPairName": String,
    "RamRoleName": String,
    "AutoReleaseTime": String,
    "SpotStrategy": String,
    "SpotPriceLimit": String,
    "SecurityEnhancementStrategy": String,
    "DiskMappings": List,
    "NetworkInterfaces": List,
    "Tags": List,
    "TemplateTags": List
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
LaunchTemplateName	String	Yes	No	The name of the instance launch template.	• The name must be 2 to 128 characters in length. • It must start with a letter but cannot start with `http://` or `https://`. • It can contain letters, digits, colons (:), underscores (_), and hyphens (-).
VersionDescription	String	No	No	The description of the version of the instance launch template.	• The description must be 2 to 128 characters in length. • It must start with a letter but cannot start with `http://` or `https://`.
ImageId	String	No	No	The ID of the image.	None
InstanceType	String	No	No	The type of the instance.	None
SecurityGroupId	String	No	No	The ID of the security group.	None
NetworkType	String	No	No	The network type of the instance.	Valid values: • classic • vpc
VSwitchId	String	No	No	The ID of the VSwitch. You must specify this parameter when you create an instance in a VPC.	None
InstanceName	String	No	No	The name of the instance.	• The name must be 2 to 128 characters in length. • It must start with a letter but cannot start with `http://` or `https://`.
Description	String	No	No	The description of the instance.	• The description must be 2 to 128 characters in length. • It must start with a letter but cannot start with `http://` or `https://`.

Property	Type	Required	Editable	Description	Constraint
InternetMaxBandwidthIn	Integer	No	No	Maximum inbound bandwidth from the Internet.	Valid values: 1 to 200. Unit: Mbit/s.
InternetMaxBandwidthOut	Integer	No	No	Maximum outbound bandwidth to the Internet.	Valid values: 0 to 100. Unit: Mbit/s.

HostName	String	No	No	The hostname of the instance.	The name cannot start or end with a period (.) or a hyphen (-). It cannot contain consecutive periods (.) or hyphens (-).
					• For Windows instances:
					  ◦ The name must be 2 to 15 characters in length and can contain letters, digits, and hyphens (-).
					  ◦ It cannot only contain digits.
					• For other instances such as Linux instances:
					  ◦ The name must be 2 to 64 characters in length and can contain letters, digits, and hyphens (-).

ZoneId	String	No	No	The ID of the zone where the instance resides.	None
SystemDiskCategory	String	No	No	The category of the system disk.	Valid values: • cloud: the basic disk • cloud_efficiency: the ultra disk • cloud_ssd: the standard SSD • ephemeral_ssd: the local SSD
SystemDiskSize	Number	No	No	The size of the system disk.	Valid values: 20 to 500. Unit: GiB.

Property	Type	Required	Editable	Description	Constraint
SystemDiskDiskName	String	No	No	The name of the system disk.	• The name must be 2 to 128 characters in length and can contain letters, digits, colons (:), underscores (_), and hyphens (-). • It must start with a letter but cannot start with `http://` or `https://`.
SystemDiskDescription	String	No	No	The description of the system disk.	The description must be 2 to 256 characters in length and cannot start with `http://` or `https://`.
IoOptimized	String	No	No	Specifies whether the instance is I/O optimized.	Valid values: • none • optimized
InternetChargeType	String	No	No	The billing method for network usage.	Valid values: • PayByBandwidth • PayByTraffic
UserData	String	No	No	The user data of the instance.	The data must be encoded in Base64. The maximum size of the raw data is 16 KB.
KeyPairName	String	No	No	The AccessKey pair name.	• For Windows instances, this parameter is ignored and is empty by default. The Password parameter takes effect even if the KeyPairName parameter is specified. • For Linux instances, the username and password authentication method is disabled by default.
RamRoleName	String	No	No	The RAM role name of the instance.	None

Property	Type	Required	Editable	Description	Constraint
AutoReleaseTime	String	No	No	The time scheduled for the instance to be automatically released.	Specify the time in the ISO 8601 standard in the `yyyy-MM-ddTHH:mm:ssZ` format. The time must be in UTC.
SpotStrategy	String	No	No	The preemption policy for pay-as-you-go instances.	This parameter takes effect only when the InstanceChargeType parameter is set to PostPaid. Valid values: • NoSpot: The instance is created as a regular pay-as-you-go instance. • SpotWithPriceLimit: The instance to be created is a preemptible instance with a user-defined maximum hourly price. • SpotAsPriceGo: The instance to be created is a preemptible instance whose price is based on the market price at the time of purchase.
SpotPriceLimit	String	No	No	The maximum hourly price of the instance.	A maximum of three decimal places can be specified.
SecurityEnhancementStrategy	String	No	No	Specifies whether to enable security hardening.	Valid values: • Active: enables security hardening. • Deactive: disables security hardening.
DiskMappings	List	No	No	The list of data disks.	A maximum of 16 data disks can be specified.
NetworkInterfaces	List	No	No	The list of elastic network interfaces (ENIs).	A maximum of eight ENIs can be specified.
Tags	List	No	No	The tags of the instance, security group, disks, and ENIs.	A maximum of 20 tags can be specified.
TemplateTags	List	No	No	The tags of the launch template.	A maximum of 20 tags can be specified.

DiskMappings syntax

```
"DiskMappings": [
  {
    "Category": String,
    "DiskName": String,
    "Description": String,
    "SnapshotId": String,
    "Size": String,
    "Encrypted": String,
    "DeleteWithInstance": String
  }
]
```

DiskMappings properties

Property	Type	Required	Editable	Description	Constraint
Category	String	No	No	The category of the data disk.	Valid values: • cloud: the basic disk • cloud_efficiency: the ultra disk • cloud_ssd: the standard SSD • ephemeral_ssd: the local SSD
DiskName	String	No	No	The name of the data disk.	• The name must be 2 to 128 characters in length. • It must start with a letter but cannot start with `http://` or `https://`. • It can contain letters, digits, colons (:), underscores (_), and hyphens (-).
Description	String	No	No	The description of the data disk.	The description must be 2 to 256 characters in length and cannot start with `http://` or `https://`.
SnapshotId	String	No	No	The ID of the snapshot used to create the data disk.	None

Property	Type	Required	Editable	Description	Constraint
Size	String	No	No	The size of the system disk.	• Valid values when the Category parameter is set to cloud: 5 to 2000. • Valid values when the Category parameter is set to cloud_efficiency: 20 to 32768. • Valid values when the Category parameter is set to cloud_ssd: 20 to 32768. • Valid values when the Category parameter is set to ephemeral_ssd: 5 to 800. Unit: GiB.
Encrypted	Boolean	No	No	Specifies whether to encrypt the data disk.	None
DeleteWithInstance	Boolean	No	No	Specifies whether to release the data disk when the instance is released.	None

NetworkInterfaces syntax

```
"NetworkInterfaces": [
  {
    "PrimaryIpAddress": String,
    "VSwitchId": String,
    "SecurityGroupId": String,
    "NetworkInterfaceName": String,
    "Description": String
  }
]
```

NetworkInterfaces properties

Property	Type	Required	Editable	Description	Constraint
PrimaryIpAddress	String	No	No	The primary private IP address of the ENI.	None
VSwitchId	String	No	No	The ID of the VSwitch to which the ENI belongs.	None
SecurityGroupId	String	No	No	The ID of the security group to which the ENI belongs.	None

Property	Type	Required	Editable	Description	Constraint
NetworkInterface Name	String	No	No	The name of the ENI.	None
Description	String	No	No	The description of the ENI.	The description must be 2 to 256 characters in length and cannot start with `http://` or `https://`.

Tags syntax

```
"Tags": [
  {
    "Value": String,
    "Key": String
  }
]
```

Tags properties

Property	Type	Required	Editable	Description	Constraint
key	String	Yes	No	None	None
value	String	No	No	None	None

TemplateTags syntax

```
"TemplateTags": [
  {
    "Value": String,
    "Key": String
  }
]
```

TemplateTags properties

Property	Type	Required	Editable	Description	Constraint
key	String	Yes	No	None	None
value	String	No	No	None	None

Response parameters

Fn::GetAtt

- LaunchTemplateId: the ID of the instance launch template.
- LaunchTemplateName: the name of the instance launch template.
- DefaultVersionNumber: the default version number of the instance launch template.
- LatestVersionNumber: the latest version number of the instance launch template.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Template1": {
      "Type": "ALIYUN::ECS::LaunchTemplate",
      "Properties": {
        "LaunchTemplateName": "MyTemplate",
        "VersionDescription": "Launch template with all properties set",
        "ImageId": "m-2ze9uqi7wo61hwep****",
        "InstanceType": "ecs.n4.small",
        "SecurityGroupId": "sg-2ze8yxgempcdsq3i****",
        "NetworkType": "vpc",
        "VSwitchId": "vsw-2zei67xd9nhcqxzec****",
        "InstanceName": "InstanceName",
        "Description": "Description of template",
        "InternetMaxBandwidthIn": 100,
        "InternetMaxBandwidthOut": 200,
        "HostName": "tttinfo",
        "ZoneId": "cn-beijing-a",
        "SystemDiskCategory": "cloud_ssd",
        "SystemDiskSize": "40",
        "SystemDiskDiskName": "TheSystemDiskName",
        "SystemDiskDescription": "The system disk description",
        "IoOptimized": "optimized",
        "InternetChargeType": "PayByBandwidth",
        "UserData": "dGhpcyBpcyBhIHVzZXIgZGF0YSBleG1h****",
        "KeyPairName": "ThisIsKeyPair",
        "RamRoleName": "ThisIsRamRole",
        "AutoReleaseTime": "2050-10-01T00:00:00Z",
        "SpotStrategy": "SpotWithPriceLimit",
        "SpotPriceLimit": "100.001",
        "SecurityEnhancementStrategy": "Active",
        "DiskMappings": [
          {
            "Category": "cloud_ssd",
            "Size": 40,
            "SnapshotId": "s-2ze1fr2bipove27b****",
            "Encrypted": true,
            "DiskName": "dataDisk1",
            "Description": "I am data disk 1",
            "DeleteWithInstance": true
      },
      {
            "Category": "cloud_efficiency",
            "Size": 20,
            "SnapshotId": "s-2ze4k0w8b33mlsqu****",
            "Encrypted": false,
            "DiskName": "dataDisk2",
            "Description": "I am data disk 2",
            "DeleteWithInstance": true
        }
      ],
      "NetworkInterfaces": [
        {
            "PrimaryIpAddress": "10.10.1.1",
            "VSwitchId": "vsw-2zetgeiqlemyok9z5****",
            "SecurityGroupId": "sg-2ze8yxgempcdsq3i****",
            "NetworkInterfaceName": "my-eni1",
            "Description": "My eni 1"
```

```
        },
      ],
      "Tags": [
           {
             "Key": "key1",
             "Value": "value1"
           },
           {
             "Key": "key2",
             "Value": "value2"
           }
         ],
      "TemplateTags": [
        {
             "Key": "templateKey1",
             "Value": "templateValue1"
              },
            {
             "Key": "templateKey2",
             "Value": "templateValue2"
            }
        ]
        }
      }
  },
  "Outputs": {
      "LaunchTemplateId": {
          "Value": {"Fn::GetAtt": ["Template1","LaunchTemplateId"]}
    },
   "LaunchTemplateName": {
        "Value": {"Fn::GetAtt": ["Template1","LaunchTemplateName"]}
    },
   "DefaultVersionNumber": {
          "Value": {"Fn::GetAtt": ["Template1","DefaultVersionNumber"]}
    },
   "LatestVersionNumber": {
          "Value": {"Fn::GetAtt": ["Template1","LatestVersionNumber"]}
    }
  }
}
```

5.1.5.1.16. ALIYUN::ECS::NatGateway

ALIYUN::ECS::NatGateway is used to create a NAT Gateway for a VPC.

Statement

```
{
  "Type": "ALIYUN::ECS::NatGateway",
  "Properties": {
    "DeletionProtection": Boolean,
    "VpcId": String,
    "Description": String,
    "NatGatewayName": String,
    "VSwitchId": String,
    "DeletionForce": Boolean,
    "Spec": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
VpcId	String	Yse	No	The ID of the VPC that you want to create NAT Gateway.	None
VSwitchId	String	Yse	No	The ID of the vSwitch in the specified VPC.	None
Description	String	Erased	Released	The description of the NAT Gateway.	The description must be 2 to 256 characters in length.
NatGatewayName	String	Erased	Released	The name of the NAT Gateway.	The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-). It must start with a letter.
Spec	String	Erased	Released	The type of the NAT Gateway.	Valid values: Small,Middle, and Large.
DeletionProtection	Boolean	Erased	Released	Indicates whether deletion protection is enabled. Default value: false.	None

Parameter	Type	Required	Editable	Description	Constraint
DeletionForce	Boolean	Erased	Released	Specifies whether to forcibly delete SNAT and DNAT entries in the Gateway and unbind EIP from the NAT gateway. Default value: false.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- ForwardTableId: the ID of the port forwarding table.
- The ID of the SNatTableId:SNat source network address translation table.
- NatGatewayId: the unique ID of the Nat gateway.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "NatGateway": {
      "Type": "ALIYUN::ECS::NatGateway",
      "Properties": {
        "NatGatewayName": "nat_gateway_1",
        "Description": "my nat gateway",
        "VpcId": "vpc-25o8s****",
        "VSwitchId": "vsw-25rc1****",
        "Spec": "Small"
      }
    }
  },
  "Outputs": {
    "NatGatewayId": {
         "Value": {"Fn::GetAttr": ["NatGateway","NatGatewayId"]}
    },
    "ForwardTableId": {
         "Value": {"Fn::GetAttr": ["NatGateway","ForwardTableId"]}
    },
    "SNatTableId": {
         "Value": {"Fn::GetAttr": ["NatGateway", "SNatTableId"]}
    }
  }
}
```

5.1.5.1.17. ALIYUN::ECS::NetworkInterface

ALIYUN::ECS::NetworkInterface is used to create an elastic network interface (ENI).

Statement

```
{
  "Type": "ALIYUN::ECS::NetworkInterface",
  "Properties": {
    "Description": String,
    "SecurityGroupId": String,
    "PrimaryIpAddress": String,
    "ResourceGroupId": String,
    "VSwitchId": String,
    "NetworkInterfaceName": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None
SecurityGroupId	String	No	Yes	The ID of the security group to which the instance belongs. The security group and the instance must be in the same VPC.	None
VSwitchId	String	No	No	The ID of the VSwitch in the VPC.	None
Description	String	Yes	True	The description of the ENI. It can contain 2 to 256 English letters or Chinese character. It cannot start with `http://` and `https://` the beginning. This parameter is empty by default.	None

Parameter	Type	Required	Editable	Description	Constraint
NetworkInterfaceName	String	Yes	True	The name of the ENI. The name must be 2 to 128 characters in length Must start with an uppercase or lowercase letter, and cannot start with `http://` and `https://` the beginning. It can contain letters, digits, colons (:), underscores (_), and hyphens (-). Default value: null.	None
PrimaryIpAddress	String	Yes	Released	The primary private IP address of the ENI. The specified IP address must be available within the CIDR block of the VSwitch. If this parameter is not specified, an available IP address in the VSwitch CIDR block is assigned at random.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- NetworkInterfaceId: the ID of the ENI.
- The MAC address of the MacAddress: Elastic Network Interface.
- The private IP address of the PrivateIpAddress: Elastic Network Interface.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "Description": {
      "Type": "String",
      "Description": "Description of your ENI. It is a string of [2, 256] English or Chinese characte
```

```
rs."
    },
    "SecurityGroupId": {
      "Type": "String",
      "Description": "The ID of the security group that the ENI joins. The security group and the ENI
must be in a same VPC."
    },
    "VSwitchId": {
      "Type": "String",
      "Description": "VSwitch ID of the specified VPC. Specifies the switch ID for the VPC."
    },
    "NetworkInterfaceName": {
      "Type": "String",
      "Description": "Name of your ENI. It is a string of [2, 128]  Chinese or English characters. It
must begin with a letter and can contain numbers, underscores (_), colons (:), or hyphens (-)."
    },
    "PrimaryIpAddress": {
      "Type": "String",
      "Description": "The primary private IP address of the ENI.  The specified IP address must have
the same Host ID as the VSwitch. If no IP addresses are specified, a random network ID is assigned fo
r the ENI."
    }
  },
  "Resources": {
    "EniInstance": {
      "Type": "ALIYUN::ECS::NetworkInterface",
      "Properties": {
        "Description": {
          "Ref": "Description"
        },
        "SecurityGroupId": {
          "Ref": "SecurityGroupId"
        },
        "VSwitchId": {
          "Ref": "VSwitchId"
        },
        "NetworkInterfaceName": {
          "Ref": "NetworkInterfaceName"
        },
        "PrimaryIpAddress": {
          "Ref": "PrimaryIpAddress"
        }
      }
    }
  },
  "Outputs": {
    "NetworkInterfaceId": {
      "Description": "ID of your Network Interface.",
      "Value": {
        "Fn::GetAtt": [
          "EniInstance",
          "NetworkInterfaceId"
        ]
      }
    }
  }
}
```

5.1.5.1.18. ALIYUN::ECS::NetworkInterfaceAttachment

ALIYUN::ECS::NetworkInterfaceAttachment is used to attach an elastic network interface (ENI) to an instance in a VPC.

Statement

```
{
  "Type": "ALIYUN::ECS::NetworkInterfaceAttachment",
  "Properties": {
    "InstanceId": String,
    "NetworkInterfaceId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
InstanceId	String	No	No	The ID of the RDS instance.	None
NetworkInterfaceId	String	No	No	The IDs of the elastic network interfaces (ENIs).	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

NetworkInterfaceId: the ID of the ENI.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "InstanceId": {
      "Type": "String",
      "Description": "ECS instance id"
    },
    "NetworkInterfaceId": {
      "Type": "String",
      "Description": "Network interface id"
    }
  },
  "Resources": {
    "EniAttachment": {
      "Type": "ALIYUN::ECS::NetworkInterfaceAttachment",
      "Properties": {
        "InstanceId": {
          "Ref": "InstanceId"
        },
        "NetworkInterfaceId": {
          "Ref": "NetworkInterfaceId"
        }
      }
    }
  },
  "Outputs": {
    "NetworkInterfaceId": {
      "Description": "ID of your Network Interface.",
      "Value": {
        "Fn::GetAtt": [
          "EniAttachment",
          "NetworkInterfaceId"
        ]
      }
    }
  }
}
```

5.1.5.1.19. ALIYUN::ECS::NetworkInterfacePermission

ALIYUN::ECS::NetworkInterfacePermission is used to grant an account the permission to attach an elastic network interface (ENI) to an instance.

Syntax

```
{
  "Type": "ALIYUN::ECS::NetworkInterfacePermission",
  "Properties": {
    "NetworkInterfaceId": String,
    "AccountId": String,
    "Permission": String
  }
}
```

Properties

Name	Type	Required	Editable	Description	Validity
NetworkInterfaceId	String	Yes	No	The ID of the ENI.	None
AccountId	String	Yes	No	The ID of the account.	None
Permission	String	Yes	No	The permission granted to the account.	None

Response parameters

Fn::GetAtt

- NetworkInterfacePermissionId: the ID of the ENI permission.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "AccountId": {
      "Type": "String",
      "Description": "the account id"
    },
    "Permission": {
      "Type": "String",
      "Description": "the permission",
      "Default": "InstanceAttach"
    },
    "NetworkInterfaceId": {
      "Type": "String",
      "Description": "Network interface id"
    }
  },
  "Resources": {
    "EniPermission": {
      "Type": "ALIYUN::ECS::NetworkInterfacePermission",
      "Properties": {
        "AccountId": {
          "Ref": "AccountId"
        },
        "Permission": {
          "Ref": "Permission"
        },
        "NetworkInterfaceId": {
          "Ref": "NetworkInterfaceId"
        }
      }
    }
  },
  "Outputs": {
    "NetworkInterfacePermissionId": {
      "Description": "the network interface permission id",
      "Value": {
        "Fn::GetAtt": [
          "EniPermission",
          "NetworkInterfacePermissionId"
        ]
      }
    }
  }
}
```

5.1.5.1.20. ALIYUN::ECS::Route

ALIYUN::ECS::Route is used to create a custom route.

Syntax

```
{
  "Type": "ALIYUN::ECS::Route",
  "Properties": {
    "DestinationCidrBlock": String,
    "RouteTableId": String,
    "NextHopId": String,
    "NextHopType": String,
    "RouteId": String,
    "NextHopList": List
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
DestinationCidrBlock	String	Yes	No	The destination Classless Inter-Domain Routing (CIDR) block of the route entry.	None
RouteTableId	String	Yes	No	The ID of the route table.	None
NextHopId	String	No	No	The ID of the next-hop instance of the route entry.	The route is a non-ECMP route.
RouteId	String	Yes	No	The ID of the route.	None
NextHopType	String	No	No	The type of the next hop.	Default value: Instance. Valid values: • Instance • Tunnel • HaVip • RouterInterface

Property	Type	Required	Editable	Description	Constraint
NextHopList	List	No	No	The list of next hops of the route entry.	You must specify the NextHopType and NextHopId parameters to specify the next hops. • If you specify the NextHopList parameter, the route is an ECMP route. The list contains two to four next hops of the ECMP route entry. Note The NextHopList parameter can be specified only when the route entry belongs to a VRouter. In addition, the next hops must be the router interfaces pointing to the connected VBRs. • If you do not specify the NextHopList parameter, the route is a non-ECMP route.

NextHopList syntax

```
"NextHopList": [
  {
    "NextHopId": String,
    "NextHopType": String
  }
]
```

NextHopList properties

Property	Type	Required	Editable	Description	Constraint
NextHopId	String	Yes	No	The ID of the next-hop instance of the route entry.	None

Property	Type	Required	Editable	Description	Constraint
NextHopType	String	No	No	The type of the next hop.	Default value: RouterInterface. Valid values: • Instance • Tunnel • HaVip • RouterInterface

Response parameters

Fn::GetAtt

None

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ECSRoute": {
      "Type": "ALIYUN::ECS::Route",
      "Properties": {
 "RouteId": "vrt-25mz0****",
        "RouteTableId": "vtb-25oud****",
        "DestinationCidrBlock": "172.16.XX.XX/24",
        "NextHopId": "i-25xzy****"
      }
    }
  }
}
```

5.1.5.1.21. ALIYUN::ECS::SNatEntry

ALIYUN::ECS::SNatEntry is used to configure a NAT Gateway table in a source network address translation.

Statement

```
{
  "Type": "ALIYUN::ECS::SNatEntry",
  "Properties": {
    "SNatTableId": String,
    "SNatIp": String,
    "SourceVSwitchId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
SNatTableId	String	Retained	Yes	The ID source network address translation table.	None

Parameter	Type	Required	Editable	Description	Constraint
SNatIp	String	Retained	Yes	The public IP address used to source network address translation.	The public IP address must be NAT Gateway in the bandwidth plan. It cannot exist in both the forwarding table and the SNAT table.
SourceVSwitchId	String	Retained	Yes	The ID of the VSwitch that accesses the Internet through the SNAT function of NAT Gateway.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

SNatEntryId: the table entry ID in the source network address translation table.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SNatEntry": {
      "Type": "ALIYUN::ECS::SNatEntry",
      "Properties": {
        "SNatTableId": "stb-3er41****",
        "SourceVSwitchId": "vsw-25rc1****",
        "SNatIp": "101.201.XX.XX"
      }
    }
  },
  "Outputs": {
    "SNatEntryId": {
         "Value": {"Fn::GetAttr": ["SNatEntry","SNatEntryId"]}
    }
}
```

5.1.5.1.22. ALIYUN::ECS::SecurityGroup

ALIYUN::ECS::SecurityGroup is used to create a security group.

Statement

```
{
  "Type": "ALIYUN::ECS::SecurityGroup",
  "Properties": {
    "VpcId": String,
    "Description": String,
    "SecurityGroupName": String,
    "Tags": List,
    "SecurityGroupEgress": List,
    "SecurityGroupIngress": List,
    "ResourceGroupId": String,
    "SecurityGroupType": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None
VpcId	String	Yes	Released	The ID of the VPC.	None
Description	String	Yes	Released	The description of the new security group.	The description must be 2 to 256 characters in length.
Tags	List	Erased	Released	The tags of the security group.	A maximum of 20 tags can be specified.

Parameter	Type	Required	Editable	Description	Constraint
SecurityGroupName	String	Yes	Released	The name of the new security group.	Default value: empty. • The name must be 2 to 128 characters in length • Must start with an uppercase or lowercase letter, and cannot start with `http://` and `https://` the beginning. • It must start with a letter and cannot start with http:// or https://.
SecurityGroupEgress	List	Erased	Released	The outbound access rules of the security group.	None
SecurityGroupIngress	List	Erased	Released	The inbound access rules of the security group.	None
SecurityGroupType	String	Yes	Released	The type of the new security group.	Valid values: • normal (basic security group) • enterprise (Advanced Security Group)

Tags syntax

```
"Tags": [
  {
    "Value" : String,
    "Key" : String
  }
]
```

Tags properties

Parameter	Type	Required	Editable	Description	Constraint
Key	String	No	No	None	None
Value	String	Yes	Released	None	None

SecurityGroupEgress syntax

```
"SecurityGroupEgress": [
  {
    "Description": String,
    "PortRange": String,
    "SecurityGroupId": String,
    "NicType": String,
    "Priority": Integer,
    "DestGroupId": String,
    "DestCidrIp": String,
    "Policy": String,
    "IpProtocol": String,
    "DestGroupOwnerAccount": String,
    "DestGroupOwnerId": String,
    "Ipv6DestCidrIp": String
  }
]
```

SecurityGroupEgress properties

Parameter	Type	Required	Editable	Description	Constraint
Description	String	Yes	Released	The description of the security group rule.	The description must be 1 to 512 characters in length.
DestGroupOwnerId	String	Yes	Released	The ID of the Alibaba Cloud account that owns the destination security group. This parameter is used to grant the current security group access to security groups in another Alibaba Cloud account.	If neither the DestGroupOwnerId parameter nor the DestGroupOwnerAccount parameter is specified, the current security group is granted access to other security groups in the same Alibaba Cloud account. If the DestCidrIp parameter is specified, the DestGroupOwnerId parameter is ignored.

Parameter	Type	Required	Editable	Description	Constraint
IpProtocol	String	No	No	The Internet protocol over which the listener will forward requests.	Valid values: • TCP • udp • icmp • GRE • All A value of all specifies that all the four protocols are supported.

Parameter	Type	Required	Editable	Description	Constraint
PortRange	String	Yes	Released	The range of port numbers corresponding to the Internet protocol.	The range of destination ports corresponding to the transport layer protocol. Valid values: • When the IpProtocol parameter is set to tcp or udp, the port number range is 1 to 65535. Separate the starting port and the ending port with a forward slash (/). Correct example: 1/200. Incorrect example: 200/1. • When the IpProtocol parameter is set to icmp, the port number range is -1/-1, indicating that all ports are available. • When the IpProtocol parameter is set to gre, the port number range is -1/-1, indicating that all ports are available. • When the IpProtocol parameter is set to all, the port number range is -1/-1.

Parameter	Type	Required	Editable	Description	Constraint
SecurityGroupId	String	Yes	Released	The ID of the security group for which to create the outbound access rules.	None
NicType	String	Yes	Released	The network type of the instance. Valid values:	Valid values: • internet • intranet Default value: internet.
Priority	String	Optional	Released	The priority of the authorization policy.	Valid values: 1 to 100. Default value: 1
DestGroupId	String	Yes	Released	The ID of the destination security group within the same region.	You must specify either the DestGroupId parameter or the DestCidrIp parameter. If both parameters are specified, the system authorizes the destination CIDR block specified by the DestCidrIp parameter. If the DestGroupId parameter is specified, but the DestCidrIp parameter is not, you must set the NicType parameter to intranet.

Parameter	Type	Required	Editable	Description	Constraint
DestCidrIp	String	Yes	Released	The source IPv4 CIDR block.	The value must be in CIDR format. The default value is 0.0.0.0/0, indicating that access from any IP addresses is allowed. Examples of other supported formats include 10.159.XX.XX/12. Only IPv4 addresses are supported.
Policy	String	Yes	Released	The authorization policy.	Valid values: • accept: grants access • drop: denies access Default value: accept.
DestGroupOwnerAccount	String	Yes	Released	The Alibaba Cloud account of the destination security group when you grant security group permissions across accounts.	None
Ipv6DestCidrIp	String	Yes	Released	The destination IPv6 CIDR block.	IPv6 addresses in CIDR format are supported. You can only specify the IP addresses for ECS instances in VPCs.

SecurityGroupIngress syntax

```
"SecurityGroupIngress": [
  {
    "SourceGroupOwnerId": String,
    "SourceGroupOwnerAccount": String,
    "Description": String,
    "PortRange": String,
    "SecurityGroupId": String,
    "NicType": String,
    "Ipv6SourceCidrIp": String,
    "Priority": Integer,
    "SourceGroupId": String,
    "Policy": String,
    "IpProtocol": String,
    "SourcePortRange": String,
    "SourceCidrIp": String
  }
]
```

SecurityGroupIngress properties

Parameter	Type	Required	Editable	Description	Constraint
SourceGroupOwnerId	String	Yes	Released	The ID of the Alibaba Cloud account that owns the source security group.	None
Description	String	Yes	Released	The description of the security group rule.	The description must be 1 to 512 characters in length.
IpProtocol	String	No	No	The Internet protocol over which the listener will forward requests.	Valid values: tcp, udp, icmp, gre, and all. A value of all specifies that all the four protocols are supported.

Parameter	Type	Required	Editable	Description	Constraint
PortRange	String	Yes	Released	The range of port numbers corresponding to the Internet protocol.	The range of destination ports corresponding to the transport layer protocol. Valid values: • When the IpProtocol parameter is set to tcp or udp, the port number range is 1 to 65535. Separate the starting port and the ending port with a forward slash (/). Correct example: 1/200. Incorrect example: 200/1. • When the IpProtocol parameter is set to icmp, the port number range is -1/-1, indicating that all ports are available. • When the IpProtocol parameter is set to gre, the port number range is -1/-1, indicating that all ports are available. • When the IpProtocol parameter is set to all, the port number range is -1/-1, indicating that all ports are available.

Parameter	Type	Required	Editable	Description	Constraint
SourceGroupId	String	Yes	Released	The ID of the source security group within the same region.	You must specify either the SourceGroupId parameter or the SourceCidrIp parameter. If both parameters are specified, the system authorizes the source CIDR block specified by the SourceCidrIp parameter. If the SourceGroupId parameter is specified, but the SourceCidrIp parameter is not, you must set the NicType parameter to intranet.
SecurityGroupId	String	Yes	Released	The ID of the security group for which you want to create the inbound access rule.	None
NicType	String	Yes	Released	The network type of the instance. Valid values:	Valid values: • internet • intranet Default value: internet.
SourceGroupOwnerAccount	String	Yes	Released	The Alibaba Cloud account of the destination security group when you grant security group permissions across accounts.	None
Priority	String	Optional	Released	The priority of the authorization policy.	Valid values: 1 to 100. Default value: 1

Parameter	Type	Required	Editable	Description	Constraint
SourceCidrIp	String	Yes	Released	The source IPv4 CIDR block.	The value must be in CIDR format. The default value is 0.0.0.0/0, indicating that access from any IP addresses is allowed. Examples of other supported formats include 10.159.XX.XX/12. Only IPv4 CIDR blocks are supported.
Policy	String	Yes	Released	The authorization policy.	Valid values: • accept: accepts the request. • drop: access is denied. Default value: accept.

Parameter	Type	Required	Editable	Description	Constraint
SourcePortRange	String	Yes	Released	The range of source ports relevant to transport layer protocols.	Valid values: • When the IpProtocol parameter is set to tcp or udp, the port number range is 1 to 65535. Separate the starting port and the ending port with a forward slash (/). Correct example: 1/200. Incorrect example: 200/1. • When the IpProtocol parameter is set to icmp, the port number range is -1/-1, indicating that all values are valid. • When the IpProtocol parameter is set to gre, the port number range is -1/-1, indicating that all ports are available. • The IpProtocol value is all: -1/-1.
Ipv6SourceCidrIp	String	Yes	Released	The source IPv6 CIDR block. IPv6 addresses in CIDR format are supported.	You can only specify the IP addresses of ECS instances in VPCs.

Response parameters

Fn::GetAtt

SecurityGroupId: the ID of the new security group.

Sample request

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Resources" : {
    "SG": {
      "Type": "ALIYUN::ECS::SecurityGroup",
      "Properties": {
        "SecurityGroupName": {
          "Ref": "SecurityGroupName"
        },
        "SecurityGroupIngress": [
          {
            "SourceCidrIp": "0.0.0.0/0",
            "IpProtocol": "all",
            "NicType": "internet",
            "PortRange": "-1/-1",
            "Priority": 1
          },
          {
            "SourceCidrIp": "0.0.0.0/0",
            "IpProtocol": "all",
            "NicType": "intranet",
            "PortRange": "-1/-1",
            "Priority": 1
          }
        ],
        "SecurityGroupEgress": [
          {
            "IpProtocol": "all",
            "DestCidrIp": "0.0.0.0/0",
            "NicType": "internet",
            "PortRange": "-1/-1",
            "Priority": 1
          },
          {
            "IpProtocol": "all",
            "DestCidrIp": "0.0.0.0/0",
            "NicType": "intranet",
            "PortRange": "-1/-1",
            "Priority": 1
          }
        ],
        "VpcId": {
          "Ref": "Vpc"
        }
      }
    }
  },
  "Outputs": {
    "SecurityGroupId": {
      "Value" : {"Fn::GetAtt": ["SG","SecurityGroupId"]}
    }
  }
}
```

5.1.5.1.23. ALIYUN::ECS::SecurityGroupClone

ALIYUN::ECS::SecurityGroupClone is used to clone a security group.

Syntax

```
{
  "Type": "ALIYUN::ECS::SecurityGroupClone",
  "Properties": {
    "DestinationRegionId": String,
    "VpcId": String,
    "Description": String,
    "SecurityGroupName": String,
    "SourceSecurityGroupId": String,
    "ResourceGroupId": String,
    "NetworkType": String,
    "SecurityGroupType": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	No	No	The ID of the resource group to which the instance belongs.	None
SourceSecurityGroupId	String	Yes	No	The ID of the source security group.	Only applicable security group rules are copied to the new security group. The security group rules are selected based on the network type of the new security group.
NetworkType	String	No	No	The network type of the new security group.	Set the value to Classic.
VpcId	String	No	No	The ID of the VPC to which the new security group belongs.	The NetworkType parameter is ignored if both the VpcId and NetworkType parameters are specified.

Property	Type	Required	Editable	Description	Constraint
Description	String	No	No	The description of the new security group.	The description must be 2 to 256 characters in length. It cannot start with http:// or https://.
SecurityGroupName	String	No	No	The name of the new security group.	This parameter is empty by default. The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It must start with a letter and cannot start with http:// or https://.
DestinationRegionId	String	No	No	The ID of the destination region where the new security group resides.	Default value: CURRENT.
SecurityGroupType	String	No	No	The type of the new security group.	Valid values: normal and enterprise. A value of normal specifies a basic security group. A value of enterprise specifies an advanced security group.

Response parameters

Fn::GetAtt

SecurityGroupId: the ID of the new security group.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SecurityGroupClone": {
      "Type": "ALIYUN::ECS::SecurityGroupClone",
      "Properties": {
        "SourceSecurityGroupId": {
          "Ref": "SourceSecurityGroupId"
```

```
        },
        "VpcId": {
          "Ref": "VpcId"
        },
        "Description": {
          "Ref": "Description"
        },
        "SecurityGroupName": {
          "Ref": "SecurityGroupName"
        },
        "DestinationRegionId": {
          "Ref": "DestinationRegionId"
        },
        "NetworkType": {
          "Ref": "NetworkType"
        }
      }
    }
  },
  "Parameters": {
    "SourceSecurityGroupId": {
      "Type": "String",
      "Description": "Source security group ID is used to copy properties to clone new security group
. If the NetworkType and VpcId is not specified, the same security group will be cloned. If NetworkTy
pe or VpcId is specified, only proper security group rules will be cloned."
    },
    "VpcId": {
      "Type": "String",
      "Description": "Physical ID of the VPC."
    },
    "Description": {
      "Type": "String",
      "Description": "Description of the security group, [2, 256] characters. Do not fill or empty, t
he default is empty."
    },
    "SecurityGroupName": {
      "Type": "String",
      "Description": "Display name of the security group, [2, 128] English or Chinese characters, mus
t start with a letter or Chinese in size, can contain numbers, '_' or '.', '-'"
    },
    "DestinationRegionId": {
      "Default": "CURRENT",
      "Type": "String",
      "Description": "Clone security group to the specified region. Default to current region."
    },
    "NetworkType": {
      "Type": "String",
      "Description": "Clone new security group as classic network type. If the VpcId is specified, th
e value will be ignored.",
      "AllowedValues": [
        "Classic"
      ]
    }
  },
  "Outputs": {
    "SecurityGroupId": {
      "Description": "Generated security group id of new security group.",
      "Value": {
        "Fn::GetAtt": [
          "SecurityGroupClone"
```

```
            "SecurityGroupId"
          ]
        }
      }
    }
}
```

5.1.5.1.24. ALIYUN::ECS::SecurityGroupEgress

ALIYUN::ECS::SecurityGroupEgress is used to create an outbound access rule for a security group.

Syntax

```
{
  "Type": "ALIYUN::ECS::SecurityGroupEgress",
  "Properties": {
    "SecurityGroupId": String,
    "IpProtocol": String,
    "PortRange": String,
    "DestGroupId": String,
    "DestGroupOwnerAccount": String,
    "DestCidrIp": String,
    "Policy": String,
    "Priority": String,
    "NicType": String,
    "Ipv6DestCidrIp": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
IpProtocol	String	Yes	No	The transport layer protocol.	Valid values: • tcp • udp • icmp • gre • all **Note** The value all indicates that all the four protocols are supported.

Property	Type	Required	Editable	Description	Constraint
PortRange	String	Yes	No	The range of destination port numbers corresponding to the transport layer protocol.	• Valid values when IpProtocol is set to tcp or udp: 1 to 65535. Separate the starting and ending port numbers with a forward slash (/). Correct example: 1/200. Incorrect example: 200/1. • Set the value to -1/-1 when IpProtocol is set to icmp. • Set the value to -1/-1 when IpProtocol is set to gre. • Set the value to -1/-1 when IpProtocol is set to all.
SecurityGroupId	String	No	No	The ID of the source security group.	None
NicType	String	No	No	The type of the network interface controller (NIC).	Default value: internet. Valid values: • internet • intranet If the DestGroupId parameter is specified, but the DestCidrIp parameter is not, set the value to intranet.
Priority	Integer	No	No	The priority of the security group rule.	Valid values: 1 to 100 Default value: 1.

Property	Type	Required	Editable	Description	Constraint
DestGroupId	String	No	No	The ID of the destination security group for which you want to set access permissions.	You must specify at least one of the DestGroupId and DestCidrIp parameters. If the DestGroupId parameter is specified, but the DestCidrIp parameter is not, set the NicType value to intranet. If both the DestGroupId and DestCidrIp parameters are specified, the DestCidrIp parameter takes precedence.
DestCidrIp	String	No	No	The destination CIDR block.	Only IPv4 CIDR blocks are supported.
Policy	String	No	No	The authorization policy.	Default value: accept. Valid values: • accept: allows access. • drop: denies access.
DestGroupOwnerAccount	String	No	No	The Alibaba Cloud account that manages the destination security group when you set a security group rule across accounts.	If you specify neither of the DestGroupOwnerAccount parameter nor the DestGroupOwnerId parameter, the access permission is configured on another security group managed by your account. If you specify the DestCidrIp parameter, the DestGroupOwnerAccount parameter is ignored.

Property	Type	Required	Editable	Description	Constraint
Description	String	No	Yes	The description of the security group rule.	The description must be 1 to 512 characters in length.
DestGroupOwnerId	String	No	No	The ID of the Alibaba Cloud account that manages the destination security group when you set a security group rule across accounts.	If you specify neither of the DestGroupOwnerAccount parameter nor the DestGroupOwnerId parameter, the access permission is configured on another security group managed by your account. If you specify the DestCidrIp parameter, the DestGroupOwnerId parameter is ignored.
Ipv6DestCidrIp	String	No	No	The destination IPv6 CIDR block.	IPv6 addresses in the CIDR format are supported. You can specify only the IP addresses of ECS instances in VPCs.

Response parameters

Fn::GetAtt

None

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SG": {
      "Type": "ALIYUN::ECS::SecurityGroupEgress",
      "Properties": {
        "SecurityGroupId": "sg-25bow****",
        "IpProtocol": "tcp",
        "PortRange": "65535/65535",
        "DestCidrIp": "0.0.0.0/0"
      }
    }
  }
}
```

5.1.5.1.25. ALIYUN::ECS::SecurityGroupIngress

ALIYUN::ECS::SecurityGroupIngress is used to create an inbound access rule for a security group.

Syntax

```
{
  "Type": "ALIYUN::ECS::SecurityGroupIngress",
  "Properties": {
    "SourceGroupOwnerId": String,
    "Description": String,
    "PortRange": String,
    "SecurityGroupId": String,
    "NicType": String,
    "Ipv6SourceCidrIp": String,
    "Priority": Integer,
    "SourceGroupId": String,
    "Policy": String,
    "IpProtocol": String,
    "SourcePortRange": String,
    "SourceCidrIp": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
IpProtocol	String	Yes	No	The Internet protocol.	Valid values: tcp, udp, icmp, gre, and all. A value of all specifies that all the four protocols are supported.

Property	Type	Required	Editable	Description	Constraint
PortRange	String	Yes	No	The range of destination ports relevant to transport layer protocols.	• When the IpProtocol parameter is set to tcp or udp, the port number range is 1 to 65535. Separate the starting port and the ending port with a forward slash (/). Correct example: 1/200. Incorrect example: 200/1. • When the IpProtocol parameter is set to icmp, the port number range is -1/-1. • When the IpProtocol parameter is set to gre, the port number range is -1/-1. • When the IpProtocol parameter is set to all, the port number range is -1/-1.

Property	Type	Required	Editable	Description	Constraint
SourceGroupId	String	No	No	The ID of the source security group for which you want to set access permissions.	You must specify at least one of the SourceGroupId and SourceCidrIp parameters. If the SourceGroupId parameter is specified, but the SourceCidrIp parameter is not, the NicType parameter must be set to intranet. If both the SourceGroupId and SourceCidrIp parameters are specified, the SourceCidrIp value is used by default.
SecurityGroupId	String	No	No	The ID of the security group for which you want to create the inbound access rule.	None
NicType	String	No	No	The network type of the instance.	Valid values: • internet • intranet Default value: internet.

Property	Type	Required	Editable	Description	Constraint
SourceGroupOwnerAccount	String	No	No	The Alibaba Cloud account that manages the source security group when you set a security group rule across accounts.	If neither the SourceGroupOwnerAccount parameter nor the SourceGroupOwnerId parameter is specified, the access permission is configured for another security group managed by your account. If the SourceCidrIp parameter is specified, this parameter is ignored.
Priority	Integer	No	No	The priority of the security group rule.	Valid values: 1 to 100. Default value: 1.
SourceCidrIp	String	No	No	The source IPv4 CIDR block.	Only IPv4 CIDR blocks are supported.
Policy	String	No	No	The access control policy.	Valid values: • accept: grants access. • drop: denies access. Default value: accept.

Property	Type	Required	Editable	Description	Constraint
SourceGroupOwnerId	String	No	No	The ID of the Alibaba Cloud account that manages the source security group when you set a security group rule across accounts.	If neither the SourceGroupOwnerId parameter nor the SourceGroupOwnerAccount parameter is specified, the access permission is configured for another security group managed by your account. If the SourceCidrIp parameter is specified, this parameter is ignored.
Description	String	No	Yes	The description of the security group rule.	The description must be 1 to 512 characters in length.

Property	Type	Required	Editable	Description	Constraint
SourcePortRange	String	No	No	The range of source ports relevant to transport layer protocols.	• When the IpProtocol parameter is set to tcp or udp, the port number range is 1 to 65535. Separate the starting port and the ending port with a forward slash (/). Correct example: 1/200. Incorrect example: 200/1. • When the IpProtocol parameter is set to icmp, the port number range is -1/-1. • When the IpProtocol parameter is set to gre, the port number range is -1/-1. • When the IpProtocol parameter is set to all, the port number range is -1/-1.
Ipv6SourceCidrIp	String	No	No	The range of source IPv6 addresses.	CIDR blocks and IPv6 addresses are supported. You can only specify the IP addresses of ECS instances in VPCs.

Response parameters

Fn::GetAtt

None

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SG": {
      "Type": "ALIYUN::ECS::SecurityGroupIngress",
      "Properties": {
        "SecurityGroupId": "sg-25bow****",
        "IpProtocol": "tcp",
        "PortRange": "65535/65535",
        "SourceCidrIp": "0.0.0.0/0"
      }
    }
  }
}
```

5.1.5.1.26. ALIYUN::ECS::Snapshot

ALIYUN::ECS::Snapshot is used to create a disk Snapshot.

Statement

```
{
  "Type": "ALIYUN::ECS::Snapshot",
  "Properties": {
    "SnapshotName": String,
    "Timeout": Integer,
    "Description": String,
    "DiskId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
DiskId	String	No	No	The ID of the disk for which you want to create the snapshot.	None

Parameter	Type	Required	Editable	Description	Constraint
SnapshotName	String	Yes	Released	The name of the snapshot.	It must be 2 to 128 characters in length. And can contain letters, digits, underscores (_), and hyphens (-). It cannot start with auto. Snapshot names starting with auto are reserved for automatic snapshots. It cannot start with `http://` or `https://`.
Timeout	String	Optional	Released	The timeout period that is specified for the snapshot creation request.	If this parameter is set, the timeout period to create a resource stack is prolonged. If the snapshot is not created within the specified time period, the entire resource stack fails to be created. You must set the timeout period according to the disk size and data amount. Valid values: 200 to 1440. Unit: minute. The default value is 200 minutes.
Description	String	Yes	Released	The description of the snapshot.	The length must be 2 to 256 characters in length. This parameter is empty by default. It cannot start with `http://` or `https://`.

Response parameters

Fn::GetAtt

SnapshotId: the ID of the snapshot.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Snapshot": {
      "Type": "ALIYUN::ECS::Snapshot",
      "Properties": {
        "DiskId": "d-2zedgvuvu8cylvrd****"
      }
    }
  },
  "Outputs": {
    "SnapshotId": {
      "Value": {
        "Fn::GetAtt": [
          "Snapshot",
          "SnapshotId"
        ]
      }
    }
  }
}
```

5.1.5.1.27. ALIYUN::ECS::SSHKeyPair

ALIYUN::ECS::SSHKeyPair is used to create or import an SSH key pair to an ECS instance.

Statement

```
{
  "Type": "ALIYUN::ECS::SSHKeyPair",
  "Properties": {
    "ResourceGroupId": String,
    "KeyPairName": String,
    "PublicKeyBody": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None

Parameter	Type	Required	Editable	Description	Constraint
KeyPairName	String	No	No	The globally unique name of the SSH key pair.	The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It cannot start with `http://` or `https://`.
PublicKeyBody	String	Yes	Released	Specifies the SSH public key to import.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- KeyPairFingerPrint: the fingerprint of the key pair. The message-digest algorithm 5 (MD5) is used based on the public key fingerprint format defined in RFC 4716.
- PrivateKeyBody: the private key of the key pair. An unencrypted RSA private key must be encoded using PEM and must be in the PKCS#8 format. The private key of a key pair can only be obtained at the time of its creation. If you import an existing public key, no private key information will be available.
- KeyPairName: the globally unique name of the SSH key pair.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SSHKeyPair": {
      "Type": "ALIYUN::ECS::SSHKeyPair",
      "Properties": {
        "KeyPairName": "ssh_key_pair_v1"
      }
    }
  },
  "Outputs": {
    "KeyPairName": {
      "Value": {
        "Fn::GetAtt": [
          "SSHKeyPair",
          "KeyPairName"
        ]
      }
    },
    "PrivateKeyBody": {
      "Value": {
        "Fn::GetAtt": [
          "SSHKeyPair",
          "PrivateKeyBody"
        ]
      }
    },
    "KeyPairFingerPrint": {
      "Value": {
        "Fn::GetAtt": [
          "SSHKeyPair",
          "KeyPairFingerPrint"
        ]
      }
    }
  }
}
```

5.1.5.1.28. ALIYUN::ECS::SSHKeyPairAttachment

ALIYUN::ECS::SSHKeyPairAttachment is used to bind an SSH key pair to an ECS instance.

Statement

```
{
  "Type": "ALIYUN::ECS::SSHKeyPairAttachment",
  "Properties": {
    "InstanceIds": List,
    "KeyPairName": String
  }
}
```

Properties

Parameter	Type	Required or Not	Editable	Description	Constraint
InstanceIds	List	Retained	Yes	The IDs of the ECS instances with which you want to associate the EIP.	Separate the IDs with a comma (,). Only Linux instances are supported.
KeyPairName	String	No	No	The name of the SSH key pair.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

None

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SSHKeyPairAttachment": {
      "Type": "ALIYUN::ECS::SSHKeyPairAttachment",
      "Properties": {
        "KeyPairName": "ssh_key_pair_v1",
        "InstanceIds": [
          'I-2zeiofnh20hj**** has been added * ',
          'I-2zebt3kfvxm2**** has two records *'
        ]
      }
    }
  }
}
```

5.1.5.1.29. ALIYUN::ECS::VPC

ALIYUN::ECS::VPC is used to create a VPC.

Statement

```
{
  "Type": "ALIYUN::ECS::VPC",
  "Properties": {
    "Description": String,
    "Ipv6CidrBlock": String,
    "EnableIpv6": Boolean,
    "ResourceGroupId": String,
    "VpcName": String,
    "CidrBlock": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None
VpcName	String	Yes	True	The name of the VPC.	• The name must be 2 to 128 characters in length • Must start with english letters or starts with a Chinese character. • It cannot start with `http://` or `https://`. • It must start with a letter and cannot start with http:// or https://.
CidrBlock	String	Yes	True	The CIDR block of the VPC.	Valid values: 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16.

Parameter	Type	Required	Editable	Description	Constraint
Description	String	Yes	True	The description of the VPC.	The description must be 2 to 256 characters in length. It cannot start with `http://` or `https://`.
Ipv6CidrBlock	String	Yes	Released	The IPv6 CIDR block of the VPC.	None
EnableIpv6	Boolean	No.	True	Specifies whether to enable an IPv6 CIDR block.	Valid values: • true • false Default value: false.

Response parameters

Fn::GetAtt

- VpcId: The VPC ID allocated by the system.
- VRouterId: the ID of the vRouter.
- RouteTableId: the ID of the routing table.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "EcsVpc": {
      "Type": "ALIYUN::ECS::VPC",
      "Properties": {
        "CidrBlock": "172.16.0.0/12",
        "VpcName": "vpc-test-del"
      }
    }
  },
  "Outputs": {
    "VpcId": {
      "Value": {
        "Fn::GetAtt": [
          "EcsVpc",
          "VpcId"
        ]
      }
    },
    "VRouterId": {
      "Value": {
        "Fn::GetAtt": [
          "EcsVpc",
          "VRouterId"
        ]
      }
    },
    "RouteTableId": {
      "Value": {
        "Fn::GetAtt": [
          "EcsVpc",
          "RouteTableId"
        ]
      }
    }
  }
}
```

5.1.5.1.30. ALIYUN::ECS::VSwitch

ALIYUN::ECS::VSwitch is used to create a VSwitch.

Statement

```
{
  "Type": "ALIYUN::ECS::VSwitch",
  "Properties": {
    "VSwitchName": String,
    "VpcId": String,
    "Description": String,
    "Ipv6CidrBlock": Integer,
    "ZoneId": String,
    "CidrBlock": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
VpcId	String	No	No	The ID of the VPC where a vSwitch is to be created	None
ZoneId	String	No	No	The ID of the zone where the instance resides.	None
VSwitchName	String	Yes	True	The name of the VSwitch.	• The name must be 2 to 128 characters in length • It must start with a letter. • Cannot `http://` or `https://` the beginning. • It must start with a letter and cannot start with http:// or https://.
CidrBlock	String	No	No	The CIDR block of the VSwitch.	The VSwitch CIDR block must be a subset of the CIDR block assigned to the VPC where the VSwitch resides and not be used by other VSwitches.
Description	String	Yes	True	The description of the vSwitch.	The description must be 2 to 256 characters in length. It cannot start with `http://` or `https://`.

Parameter	Type	Required	Editable	Description	Constraint
Ipv6CidrBlock	String	Optional	Released	The IPv6 CIDR block of the VSwitch. You can customize the last eight bits of the IPv6 CIDR block.	Valid values: 0 to 255. The value is a decimal integer. By default, the prefix of the IPv6 CIDR block of the VSwitch is set to /64.

Response parameters

Fn::GetAtt

- VSwitchId: indicates the vSwitch ID allocated by the system.
- CidrBlock: the IPv4 CIDR block of the vSwitch.
- Ipv6CidrBlock: the IPv6 CIDR block of the vSwitch.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "VpcName": {
      "Type": "String"
    },
    "VSwitch1CidrBlock": {
      "Type": "String",
      "Default": "172.16.100.0/24"
    },
    "VSwitch2CidrBlock": {
        "Type": "String",
        "Default": "172.16.80.0/24"
    }
  },
  "Resources": {
    "EcsVpc": {
      "Type": "ALIYUN::ECS::VPC",
      "Properties": {
        "CidrBlock": "172.16.0.0/12",
        "VpcName": {"Ref": "VpcName"},
      },
    },
    "VSwitch1": {
      "Type": "ALIYUN::ECS::VSwitch",
      "Properties": {
        "ZoneId": "cn-beijing-a",
        "CidrBlock": {"Ref": "VSwitch1CidrBlock"},
        "VpcId": { "Fn::GetAtt": [ "EcsVpc", "VpcId" ] },
        "VSwitchName": "create_vpc_vswitch_sg1"
      }
    },
    "VSwitch2": {
      "Type": "ALIYUN::ECS::VSwitch",
      "Properties": {
        "ZoneId": "cn-beijing-a",
        "CidrBlock": {"Ref": "VSwitch2CidrBlock"},
```

```
        "VpcId": { "Fn::GetAtt": [ "EcsVpc", "VpcId" ] },
        "VSwitchName": "create_vpc_vswitch_sg2"
      }
    },
    "SG_VSwitch1": {
      "Type": "ALIYUN::ECS::SecurityGroup",
      "Properties": {
        "SecurityGroupName": "app_mall",
        "Description": "this is created by heat",
        "VpcId": { "Fn::GetAtt": [ "EcsVpc", "VpcId" ] }
      },
      "Outputs": {
        "SecurityGroupId": {
             "Value": {"get_attr": ["SG_VSwitch1","SecurityGroupId"]}
        }
      }
    },
    "SG_VSwitch1_InRule": {
      "Type": "ALIYUN::ECS::SecurityGroupIngress",
      "Properties": {
        "SecurityGroupId": { "Fn::GetAtt": [ "SG_VSwitch1", "SecurityGroupId" ] },
        "IpProtocol": "tcp",
        "PortRange": "1/65535",
        "SourceCidrIp": {"Ref": "VSwitch2CidrBlock"}
      }
    },
    "SG_VSwitch1_OutRule": {
      "Type": "ALIYUN::ECS::SecurityGroupEgress",
      "Properties": {
        "SecurityGroupId": { "Fn::GetAtt": [ "SG_VSwitch1", "SecurityGroupId" ] },
        "IpProtocol": "tcp",
        "PortRange": "1/65535",
        "DestCidrIp": {"Ref": "VSwitch2CidrBlock"}
      }
    },
    "SG_VSwitch2": {
      "Type": "ALIYUN::ECS::SecurityGroup",
      "Properties": {
        "SecurityGroupName": "app_mall",
        "Description": "this is created by heat",
        "VpcId": { "Fn::GetAtt": [ "EcsVpc", "VpcId" ] }
      },
    },
    "SG_VSwitch2_InRule": {
      "Type": "ALIYUN::ECS::SecurityGroupIngress",
      "Properties": {
        "SecurityGroupId": { "Fn::GetAtt": [ "SG_VSwitch2", "SecurityGroupId" ] },
        "IpProtocol": "tcp",
        "PortRange": "1/65535",
        "SourceCidrIp": {"Ref": "VSwitch1CidrBlock"}
      }
    },
    "SG_VSwitch2_OutRule": {
      "Type": "ALIYUN::ECS::SecurityGroupEgress",
      "Properties": {
        "SecurityGroupId": { "Fn::GetAtt": [ "SG_VSwitch2", "SecurityGroupId" ] },
        "IpProtocol": "tcp",
        "PortRange": "1/65535",
        "DestCidrIp": {"Ref": "VSwitch1CidrBlock"}
      }
```

```
      }
    }
  }
}
```

5.1.5.2. ESS

5.1.5.2.1. ALIYUN::ESS::AlarmTask

ALIYUN::ESS::AlarmTask is used to create a metric-based alarm task.

Syntax

```
{
  "Type": "ALIYUN::ESS::AlarmTask",
  "Properties": {
    "Statistics": String,
    "Name": String,
    "EvaluationCount": Integer,
    "Period": Integer,
    "MetricType": String,
    "ComparisonOperator": String,
    "Dimensions": List,
    "ScalingGroupId": String,
    "AlarmAction": List,
    "Threshold": Number,
    "MetricName": String,
    "GroupId": Integer,
    "Description": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
Statistics	String	No	No	The method used to calculate monitoring data. The statistics must be appropriate for the metric chosen.	Valid values: Average, Minimum, and Maximum. Default value: Average.
Name	String	No	Yes	The name of the alarm rule.	None
EvaluationCount	Integer	No	No	The number of consecutive times that the threshold must be exceeded before an alarm is triggered.	Default value: 3. Minimum value: 1.
Period	Integer	No	No	The metric query period, which must be appropriate for the metric chosen. Unit: seconds.	Valid values: 60, 120, 300, and 900. Default value: 300.

Property	Type	Required	Editable	Description	Constraint
MetricType	String	No	No	The metric type.	Valid values: system and custom.
ComparisonOperator	String	No	No	The alarm comparison operator used to define a condition in the alarm rule.	Valid values: <=, <, >, and > =.
Dimensions	List	No	No	The list of instances associated with the alarm rule.	You must include at least one instance in the list.
ScalingGroupId	String	Yes	No	The ID of the scaling group.	None
AlarmAction	List	Yes	Yes	The list of alarm actions.	You must include one to five alarm actions in the list.
Threshold	Number	Yes	No	The alarm threshold, which must be a numeric value.	None
MetricName	String	Yes	No	The metric name of a service. For more information, see the metrics defined for each service.	None
GroupId	Integer	No	No	The group ID.	None
Description	String	No	Yes	The description of the alarm task.	None

Dimensions syntax

```
"Dimensions": [
  {
    "DimensionKey": String,
    "DimensionValue": String
  }
]
```

Dimensions properties

Property	Type	Required	Editable	Description	Constraint
DimensionValue	String	Yes	No	None	None
DimensionKey	String	Yes	No	None	None

Response parameters

Fn::GetAtt

AlarmTaskId: the ID of the alarm task.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "ComparisonOperator": {
      "Type": "String",
      "Description": "Comparison Operator",
      "AllowedValues": [
        ">=",
        "<=",
        ">",
        "<"
      ]
    },
    "Description": {
      "Type": "String",
      "Description": "Description"
    },
    "ScalingGroupId": {
      "Type": "String",
      "Description": "The ID of the scaling group."
    },
    "MetricType": {
      "Type": "String",
      "Description": "Metric Type",
      "AllowedValues": [
        "system",
        "custom"
      ]
    },
    "EvaluationCount": {
      "Type": "Number",
      "Description": "Evaluation Count",
      "MinValue": 1
    },
    "Period": {
      "Type": "Number",
      "Description": "Period",
      "AllowedValues": [
        60,
        120,
        300,
        900
      ]
    },
    "Dimensions": {
      "Type": "CommaDelimitedList",
      "Description": "Dimensions",
      "MinLength": 1
    },
    "Statistics": {
      "Type": "String",
      "Description": "Statistics",
      "AllowedValues": [
        "Average",
        "Minimum",
        "Maximum"
      ]
```

```
    },
    "Name": {
      "Type": "String",
      "Description": "Name"
    },
    "GroupId": {
      "Type": "Number",
      "Description": "Group Id"
    },
    "MetricName": {
      "Type": "String",
      "Description": "Metric Name"
    },
    "AlarmAction": {
      "Type": "CommaDelimitedList",
      "Description": "Alarm Actions",
      "MinLength": 1,
      "MaxLength": 5
    },
    "Threshold": {
      "Type": "Number",
      "Description": "Threshold"
    }
  },
  "Resources": {
    "AlarmTask": {
      "Type": "ALIYUN::ESS::AlarmTask",
      "Properties": {
        "ComparisonOperator": {
          "Ref": "ComparisonOperator"
        },
        "Description": {
          "Ref": "Description"
        },
        "ScalingGroupId": {
          "Ref": "ScalingGroupId"
        },
        "MetricType": {
          "Ref": "MetricType"
        },
        "EvaluationCount": {
          "Ref": "EvaluationCount"
        },
        "Period": {
          "Ref": "Period"
        },
        "Dimensions": {
          "Fn::Split": [
            ",",
            {
              "Ref": "Dimensions"
            },
            {
              "Ref": "Dimensions"
            }
          ]
        },
        "Statistics": {
          "Ref": "Statistics"
        },
```

```
        },
        "Name": {
          "Ref": "Name"
        },
        "GroupId": {
          "Ref": "GroupId"
        },
        "MetricName": {
          "Ref": "MetricName"
        },
        "AlarmAction": {
          "Fn::Split": [
            ",",
            {
              "Ref": "AlarmAction"
            },
            {
              "Ref": "AlarmAction"
            }
          ]
        },
        "Threshold": {
          "Ref": "Threshold"
        }
      }
    }
  },
  "Outputs": {
    "AlarmTaskId": {
      "Description": "The alarm task ID",
      "Value": {
        "Fn::GetAtt": [
          "AlarmTask",
          "AlarmTaskId"
        ]
      }
    }
  }
}
```

5.1.5.2.2. ALIYUN::ESS::AlarmTaskEnable

ALIYUN::ESS::AlarmTaskEnable is used to start an alarm task. You can call this operation to enable alarm tasks when the task is stopped.

Statement

```
{
  "Type": "ALIYUN::ESS::AlarmTaskEnable",
  "Properties": {
    "AlarmTaskId": String,
    "Enable": Boolean
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
AlarmTaskId	String	No	No	The ID of the monitoring task.	None
Enable	String	Retained	Yes	Specifies whether to enable the alarm task.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

None

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "Enable": {
      "Type": "Boolean",
      "Description": "Enable alarm task or not",
      "AllowedValues": [
        "True",
        "true",
        "False",
        "false"
      ]
    },
    "AlarmTaskId": {
      "Type": "String",
      "Description": "The id of alarm task."
    }
  },
  "Resources": {
    "AlarmTaskEnable": {
      "Type": "ALIYUN::ESS::AlarmTaskEnable",
      "Properties": {
        "Enable": {
          "Ref": "Enable"
        },
        "AlarmTaskId": {
          "Ref": "AlarmTaskId"
        }
      }
    }
  },
  "Outputs": {}
}
```

5.1.5.2.3. ALIYUN::ESS::LifecycleHook

ALIYUN::ESS::LifecycleHook is used to create a lifecycle hook for a scaling group.

Syntax

```
{
  "Type": "ALIYUN::ESS::LifecycleHook",
  "Properties": {
    "LifecycleHookName": String,
    "NotificationArn": String,
    "HeartbeatTimeout": Integer,
    "NotificationMetadata": String,
    "ScalingGroupId": String,
    "DefaultResult": String,
    "LifecycleTransition": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
LifecycleHookName	String	No	Yes	The name of the lifecycle hook. Each lifecycle hook name must be unique within a scaling group.	The name must be 2 to 40 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit. The default name is the ID of the lifecycle hook.
NotificationArn	String	No	Yes	The Alibaba Cloud Resource Name (ARN) of the notification target that Auto Scaling uses to notify you when an instance is in the transition state for the lifecycle hook.	This target can be either an MNS queue or an MNS topic. The format of the parameter value is acs:ess:{region}:{account-id}:{resource-relative-id}. • `region`: the region where the scaling group resides. • `account-id`: the ID of the Apsara Stack tenant account. Examples: • MNS queue: acs:ess:{region}:{account-id}:queue/{queuename} • MNS topic: acs:ess:{region}:{account-id}:topic/{topicname}

Property	Type	Required	Editable	Description	Constraint
HeartbeatTimeout	Integer	No	Yes	The waiting period before the lifecycle hook times out. When the lifecycle hook times out, the scaling group performs the action specified by the DefaultResult parameter. Unit: seconds.	Valid values: 30 to 21600. Default value: 600.
NotificationMetadata	String	No	Yes	The fixed string to include when Auto Scaling sends a notification about the wait state of a scaling activity. Auto Scaling sends the specified `NotificationMetadata` parameter value along with the notification message so that you can easily categorize notifications. The `NotificationMetadata` parameter is valid only after you set the `NotificationArn` parameter.	The parameter value cannot exceed 128 characters in length.
ScalingGroupId	String	Yes	No	The ID of the scaling group.	None

Property	Type	Required	Editable	Description	Constraint
DefaultResult	String	No	Yes	The action that the scaling group takes when the lifecycle hook times out. If the scaling group has multiple lifecycle hooks and one of them is terminated when the `DefaultResult` parameter is set to ABANDON during a `scale-in` event, the remaining lifecycle hooks in the same scaling group will also be terminated. Otherwise, the scaling activity will proceed normally after the waiting period expires and continue with the action specified by the DefaultResult parameter.	Valid values: • CONTINUE: The scaling group continues the scale-in or scale-out event. • ABANDON: The scaling group releases the created ECS instances if the scaling activity type is scale-out or removes the ECS instances to be scaled in if the scaling activity type is scale-in. Default value: CONTINUE.
LifecycleTransition	String	Yes	Yes	The type of scaling activity to which the lifecycle hook applies.	Valid values: • SCALE_OUT: scale-out events of the scaling group. • SCALE_IN: scale-in events of the scaling group.

Response parameters

Fn::GetAtt

LifecycleHookId: the ID of the lifecycle hook.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "LifecycleHookName": {
      "Type": "String",
      "Description": "The name of the lifecycle hook. Each name must be unique within a scaling group. The name must be 2 to 40 characters in length and can contain letters, numbers, Chinese characters, and special characters including underscores (_), hyphens (-) and periods (.).\nDefault value: Lifecycle Hook ID",
      "AllowedPattern": "^[a-zA-Z0-9\\u4e00-\\u9fa5][-_.a-zA-Z0-9\\u4e00-\\u9fa5]{1,63}$"
    },
    "NotificationArn": {
      "Type": "String",
      "Description": "The Alibaba Cloud Resource Name (ARN) of the notification target that Auto Scaling will use to notify you when an instance is in the transition state for the lifecycle hook. This target can be either an MNS queue or an MNS topic. The format of the parameter value is acs:ess:{region}:{account-id}:{resource-relative-id}.\nregion: the region to which the scaling group locates\naccou
```

```
nt-id: Alibaba Cloud ID\nFor example:\nMNS queue: acs:ess:{region}:{account-id}:queue/{queuename}\nMN
S topic: acs:ess:{region}:{account-id}:topic/{topicname}",
      "AllowedPattern": "^acs:ess:([a-zA-Z0-9-]+):(\\d+):(queue|topic)/([a-zA-Z0-9][a-zA-Z0-9-]{0,255
})$",
      "MaxLength": 300
    },
    "ScalingGroupId": {
      "Type": "String",
      "Description": "The ID of the scaling group."
    },
    "LifecycleTransition": {
      "Type": "String",
      "Description": "The scaling activities to which lifecycle hooks apply Value range:\n SCALE_OUT:
scale-out event\n SCALE_IN: scale-in event",
      "AllowedValues": [
        "SCALE_OUT",
        "SCALE_IN"
      ]
    },
    "HeartbeatTimeout": {
      "Type": "Number",
      "Description": "The time, in seconds, that can elapse before the lifecycle hook times out. If t
he lifecycle hook times out, the scaling group performs the default action (DefaultResult). The range
is from 30 to 21,600 seconds. The default value is 600 seconds.\nYou can prevent the lifecycle hook f
rom timing out by calling the RecordLifecycleActionHeartbeat operation. You can also terminate the li
fecycle action by calling the CompleteLifecycleAction operation.",
      "MinValue": 30,
      "MaxValue": 21600
    },
    "NotificationMetadata": {
      "Type": "String",
      "Description": "The fixed string that you want to include when Auto Scaling sends a message abo
ut the wait state of the scaling activity to the notification target. The length of the parameter can
be up to 128 characters. Auto Scaling will send the specified NotificationMetadata parameter along wi
th the notification message so that you can easily categorize your notifications. The NotificationMet
adata parameter will only take effect after you specify the NotificationArn parameter.",
      "MaxLength": 128
    },
    "DefaultResult": {
      "Type": "String",
      "Description": "The action that the scaling group takes when the lifecycle hook times out. Valu
e range:\n CONTINUE: the scaling group continues with the scale-in or scale-out process.\n ABANDON: t
he scaling group stops any remaining action of the scale-in or scale-out event.\nDefault value: CONTI
NUE\nIf the scaling group has multiple lifecycle hooks and one of them is terminated by the DefaultRe
sult=ABANDON parameter during a scale-in event (SCALE_IN), the remaining lifecycle hooks under the sa
me scaling group will also be terminated. Otherwise, the action following the wait state is the next
action, as specified in the parameter DefaultResult, after the last lifecycle event under the same sc
aling group.",
      "AllowedValues": [
        "CONTINUE",
        "ABANDON"
      ]
    }
  },
  "Resources": {
    "LifecycleHook": {
      "Type": "ALIYUN::ESS::LifecycleHook",
      "Properties": {
        "LifecycleHookName": {
```

```
            "Ref": "LifecycleHookName"
          },
          "NotificationArn": {
            "Ref": "NotificationArn"
          },
          "ScalingGroupId": {
            "Ref": "ScalingGroupId"
          },
          "LifecycleTransition": {
            "Ref": "LifecycleTransition"
          },
          "HeartbeatTimeout": {
            "Ref": "HeartbeatTimeout"
          },
          "NotificationMetadata": {
            "Ref": "NotificationMetadata"
          },
          "DefaultResult": {
            "Ref": "DefaultResult"
          }
        }
      }
    },
    "Outputs": {
      "LifecycleHookId": {
        "Description": "The lifecycle hook ID",
        "Value": {
          "Fn::GetAtt": [
            "LifecycleHook",
            "LifecycleHookId"
          ]
        }
      }
    }
}
```

5.1.5.2.4. ALIYUN::ESS::ScalingConfiguration

ALIYUN::ESS::ScalingConfiguration is used to create a scaling configuration for a scaling group.

Statement

```
{
  "Type": "ALIYUN::ESS::ScalingConfiguration",
  "Properties": {
    "PasswordInherit": Boolean,
    "DiskMappings": List,
    "RamRoleName": String,
    "IoOptimized": String,
    "InternetChargeType": String,
    "KeyPairName": String,
    "InstanceId": String,
    "InstanceTypes": List,
    "ImageId": String,
    "ResourceGroupId": String,
    "SpotStrategy": String,
    "InstanceType": String,
    "SystemDiskCategory": String,
    "SystemDiskSize": Integer,
    "SystemDiskAutoSnapshotPolicyId": String,
    "InternetMaxBandwidthOut": Integer,
    "InstanceName": String,
    "InternetMaxBandwidthIn": Integer,
    "ScalingConfigurationName": String,
    "UserData": String,
    "DeploymentSetId": String,
    "SecurityGroupId": String,
    "SpotPriceLimit": Number,
    "HpcClusterId": String,
    "ScalingGroupId": String,
    "SpotPriceLimitForInstanceType": Map,
    "TagList": List
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	True	The ID of the resource group to which the instance belongs.	None
DeploymentSetId	String	Yes	Released	The ID of the deployment set.	None
HpcClusterId	String	Yes	Released	The ID of the E-HPC cluster to which the instance belongs.	None
ScalingGroupId	String	No	No	The ID of the scaling group to which the scaling configuration belongs.	None
DiskMappings	List	No.	True	The disks to be attached to created instances.	A maximum of 16 disks can be attached to each instance.

Parameter	Type	Required	Editable	Description	Constraint
InternetChargeType	String	Yes	True	The billing method for Internet usage.	Valid values: • PayByBandwidth • PayByTraffic: pay-by-traffic Default value: PayByTraffic
InternetMaxBandwidthIn	String	Optional	Released	The maximum inbound bandwidth from the Internet.	Unit: Mbit/s. Valid values: 1 to 100. Default value: 100
InternetMaxBandwidthOut	String	No.	True	The maximum outbound bandwidth to the Internet.	Valid values: • Pay-by-bandwidth: 0 to 100. Default value: 0. • Pay-by-data-transfer: 1 to 200. This parameter is required. Unit: Mbit/s.
InstanceId	String	Yes	Released	The instance ID of the scaling configuration.	None
SystemDiskCategory	String	Yes	True	The category of the system disk.	Valid values: • cloud: indicates a basic disk. • cloud_efficiency: indicates an ultra disk. • cloud_ssd: indicates a standard SSD. • ephemeral_ssd: indicates a local SSD. • cloud_essd: enhanced SSD (ESSD) Default value: cloud for Generation I instance types that are not I/O optimized, default value: cloud_efficiency.
ImageId	String	Yes	True	The ID of the image used to start the instance. You can use a public image, a custom image, or an Alibaba Cloud Marketplace image.	None
InstanceType	String	Yes	True	The specification of the instance.	None

Parameter	Type	Required	Editable	Description	Constraint
SecurityGroupId	String	Yes	True	The ID of the security group to which the instance belongs.	None
IoOptimized	String	Yes	True	Specifies whether the created instances are I/O optimized.	Valid values: • none (non-I/O optimized) • optimized Default value: none.
ScalingConfigurationName	String	Yes	True	The name of the scaling configuration.	• The name must be 2 to 64 characters in length. It can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit. • The name of the scaling configuration must be unique within a scaling group. • If this parameter is not specified, the value of scalingconfigurationid is used.
KeyPairName	String	Yes	True	The name of the key pair that is bound to the instance.	• This parameter is ignored if the instance type is Windows and the default value is null. • If the instance type is Linux, password logon is disabled by default.
RamRoleName	String	Yes	True	The RAM role name of the instance.	You can use RAM API ListRoles you can call this operation to query the RAM role name of an instance.
SystemDiskSize	String	No.	True	The size of the system disk. Unit: GB.	Valid values: 40 to 500. Unit: GB. If a custom image is used to create a system disk, the system disk size must be larger than the size of the custom image.

Parameter	Type	Required	Editable	Description	Constraint
UserData	String	Yes	True	The user data that you pass when you create the instance.	The user can encode up to 16KB in size. You do not need to perform Base64 encoding. Special characters must be escaped with a backslash (\).
InstanceTypes	List	No.	True	The instance types from which ECS instances can be created. If you specify InstanceTypes, InstanceType is invalid.	Up to 10 instance types can be configured in a scaling configuration. The priority of each instance type is decreased in the order of its list elements. Auto Scaling creates instances in order of priority. If an instance of the highest priority type cannot be created, Auto Scaling will create an instance of the next highest priority type.
PasswordInherit	Boolean	No.	True	Specifies whether to use the preconfigured password of the specified image.	To use this parameter, ensure that a password is configured for the specified image.
TagList	List	No.	True	The tags of the instance.	Tags must be specified as key-value pairs. You can specify a maximum of five Tag groups in the format of `{"key1": "value1", "key2": "value2", ... "key5": "value5"}`. The key can contain a maximum of 64 characters. `aliyun`, `http://` or `https://` the beginning. If you use tags, the key cannot be an empty string. The value must be 0 to 128 characters in length.

Parameter	Type	Required	Editable	Description	Constraint
SpotStrategy	String	Yes	True	The preemption policy for pay-as-you-go instances.	Valid values: • NoSpot (pay-as-you-go instance) • SpotWithPriceLimit (a preemptible instance with a maximum price) • SpotAsPriceGo (the SpotAsPriceGo parameter that is set automatically based on the actual market price.) Default value: NoSpot.
InstanceName	String	Yes	True	The name of the instance created based on the current scaling configuration.	None
SpotPriceLimit	Number	No.	True	The maximum hourly price of the instance.	A maximum of three decimal places can be specified. This parameter takes effect only when the SpotStrategy parameter is set to SpotWithPriceLimit. The value of this parameter can be overwritten by the value of the SpotPriceLimitForInstanceType parameter.
SpotPriceLimitForInstanceType	Map	No.	True	Preemptible instance type and bid of the instance.	The format is `{"<instance_type_1>":<price_limit_1>, ..., {"<instance_type_10>":<price_limit_10>}`. This parameter takes effect only when the SpotStrategy parameter is set to SpotWithPriceLimit. You can set up to 10 instance groups and prices.
SystemDiskAutoSnapshotPolicyId	String	Yes	True	The ID of the automatic snapshot policy applied to the data disk.	None

DiskMappings syntax

```
"DiskMappings": [
  {
    "Category": String,
    "Device": String,
    "SnapshotId": String,
    "Size": String,
    "Encrypted": String,
    "KMSKeyId": String,
    "Description": String,
    "DiskName": String
  }
]
```

DiskMappings properties

Parameter	Type	Required	Editable	Description	Constraint
Size	String	No	No	The size of the data disk.	Valid values: • cloud: 5 to 2000 • cloud_efficiency: 20 to 32768 • cloud_ssd: 20 to 32768 • cloud_essd: 20 to 32768. • ephemeral_ssd: 5 to 800 The value of this parameter must be greater than or equal to that of the snapshot specified by SnapshotId. Unit: GiB.
Category	String	Yes	Released	The type of the data disk.	Valid values: cloud, cloud_efficiency, cloud_ssd, ephemeral_ssd, and cloud_essd. For I/O optimized instances, the default value is cloud_efficiency. For non-I/O optimized instances, the default value is cloud.
DiskName	String	Yes	Released	The name of the data disk.	The name must be 2 to 128 characters in length It can contain letters, digits, colons (:), underscores (_), and hyphens (-). It must start with a letter and cannot start with `http://` or `https://` the beginning.

Parameter	Type	Required	Editable	Description	Constraint
Description	String	Yes	Released	The description of the data disk.	The description must be 2 to 256 characters in length. Cannot `http://` or `https://` the beginning.
Device	String	Yes	Released	The device name of the data disk.	By default, the system automatically assigns a value for this parameter when the ECS instance is created. The value starts from /dev/xvdb and ends at /dev/xvdz.
SnapshotId	String	Yes	Released	The ID of the snapshot used to create the data disk.	If this parameter is specified, the Size parameter will be ignored, and the Size of the created disk will be the Size of the specified snapshot. If the snapshot was created on or before July 15, 2013, calling the snapshot is denied and InvalidSnapshot.TooOld is displayed in the response parameter.
Encrypted	String	Yes	Released	Specifies whether to encrypt the data disk.	Default value: false.
KMSKeyId	String	Yes	Released	The KMS key ID for data disk N.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

ScalingConfigurationId: the ID of the scaling configuration. This ID is a globally unique identifier (GUID) generated by the system.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ScalingConfiguration": {
      "Type": "ALIYUN::ESS::ScalingConfiguration",
      "Properties": {
        "ImageId": "ubuntu1404_64_20G_aliaegis_2015****.vhd",
        "InstanceType": "ecs.t1.small",
        "InstanceId": "i-25xhh****",
        "InternetChargeType": "PayByTraffic",
        "InternetMaxBandwidthIn": 1,
        "InternetMaxBandwidthOut": 20,
        "SystemDisk_Category": "cloud",
        "ScalingGroupId": "bwhtvpcBcKYac9fe3vd0****",
        "SecurityGroupId": "sg-25zwc****",
        "DiskMappings": [
          {
            "Size": 10
          },
          {
            "Category": "cloud",
            "Size": 10
          }
        ]
      }
    }
  },
  "Outputs": {
    "ScalingConfiguration": {
      "Value": {"get_attr": ["ScalingConfigurationId"]}
    }
  }
}
```

5.1.5.2.5. ALIYUN::ESS::ScalingGroup

ALIYUN::ESS::ScalingGroup is used to create a scaling group. A scaling group is a group of ECS instances that are dynamically scaled based on the configured scenario. A scaling group does not take effect immediately after it is created. You must use ALIYUN::ESS::ScalingGroupEnable to enable the scaling group to trigger scaling rules and execute scaling activities.

Syntax

```
{
  "Type": "ALIYUN::ESS::ScalingGroup",
  "Properties": {
    "MultiAZPolicy": String,
    "DesiredCapacity": Integer,
    "NotificationConfigurations": List,
    "ProtectedInstances": List,
    "LaunchTemplateId": String,
    "LaunchTemplateVersion": String,
    "ScalingGroupName": String,
    "VSwitchIds": List,
    "DefaultCooldown": Integer,
    "MinSize": Integer,
    "GroupDeletionProtection": Boolean,
    "MaxSize": Integer,
    "InstanceId": String,
    "VSwitchId": String,
    "LoadBalancerIds": List,
    "StandbyInstances": List,
    "RemovalPolicys": List,
    "HealthCheckType": String,
    "DBInstanceIds": List
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
MinSize	Integer	Yes	Yes	The minimum number of ECS instances in the scaling group.	Valid values: 0 to 1000. When the number of ECS instances in the scaling group is less than the MinSize value, Auto Scaling automatically creates ECS instances until the number of instances is equal to the MinSize value.

Property	Type	Required	Editable	Description	Constraint
MaxSize	Integer	Yes	Yes	The maximum number of ECS instances in the scaling group.	Valid values: 0 to 1000. When the number of ECS instances in the scaling group is greater than the MaxSize value, Auto Scaling removes ECS instances from the scaling group until the number of instances is equal to the MaxSize value.
ScalingGroupName	String	No	Yes	The display name of the scaling group.	• The name must be 2 to 40 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). • It must start with an uppercase letter, lowercase letter, or digit. • The name must be unique to an Alibaba Cloud account in a region. The default value is the ID of the scaling group.
LaunchTemplateId	String	No	Yes	The ID of the instance launch template from which the scaling group obtains launch configurations.	None

Property	Type	Required	Editable	Description	Constraint
LaunchTemplateVersion	String	No	Yes	The version of the instance launch template.	Valid values: • The fixed template version number. • Default: The default template version is always used. • Latest: The latest template version is always used.
RemovalPolicys	List	No	Yes	The list of one or more policies that are used to remove ECS instances from the scaling group.	Default value: OldestScalingConfiguration or OldestInstance. Valid values: • OldestInstance: removes the ECS instance that is added to the scaling group at the earliest point in time. • NewestInstance: removes the ECS instance that is added to the scaling group at the latest point in time. • OldestScalingConfiguration: removes the ECS instance that is created based on the earliest scaling configuration.
VSwitchId	String	No	No	The ID of the vSwitch.	None

Property	Type	Required	Editable	Description	Constraint
LoadBalancerIds	List	No	Yes	The ID of the Server Load Balancer (SLB) instance.	This value can be a JSON array that contains up to five SLB instance IDs. Separate multiple IDs with commas (,).
DefaultCooldown	Integer	No	Yes	The cooldown time after a scaling activity (adding or removing ECS instances) is executed.	• Valid values: 0 to 86400. • Unit: seconds. • Default value: 300. During the cooldown time, the scaling group executes only scaling activities that are triggered by Cloud Monitor event-triggered tasks.
DBInstanceIds	List	No	Yes	The list of one or more ApsaraDB RDS instance IDs.	This value can be a JSON array that contains up to eight ApsaraDB RDS instance IDs. Separate multiple IDs with commas (,).

Property	Type	Required	Editable	Description	Constraint
VSwitchIds	List	No	No	The list of one or more vSwitch IDs.	You can specify a maximum of five vSwitch IDs. If you specify this parameter, the VSwitchId parameter is ignored. vSwitches are sorted in descending order of priority. When an ECS instance cannot be created in the zone where the vSwitch with the highest priority resides, the system uses the vSwitch with the next highest priority to create the ECS instance.
MultiAZPolicy	String	No	No	The ECS instance scaling policy for the multi-	Valid values: • PRIORITY: ECS instances are scaled based on the specified vSwitch. When an ECS instance cannot be created in the zone where the vSwitch with the highest priority resides, the system uses the vSwitch with the next highest priority to create the ECS instance. • BALANCE: ECS instances are distributed evenly in multiple zones specified in the scaling group.

Property	Type	Required	Editable	Description	Constraint
				zone scaling group.	• COST_OPTIMIZED: ECS instances are created based on the unit price of vCPUs, from low to high. Preemptible instances are created first when preemptible instance types are specified for the scaling configuration. Pay-as-you-go instances are automatically created when no preemptible instances are available due to issues such as insufficient ECS resources.
NotificationConfigurations	List	No	Yes	The notification configurations for event and resource changes.	None
ProtectedInstances	List	No	Yes	The number of protected ECS instances in the scaling group.	Maximum value: 1000.
StandbyInstances	List	No	Yes	The number of ECS instances that are in the standby state in the scaling group.	Maximum value: 1000.
HealthCheckType	String	No	Yes	The health check type.	Valid values: • ECS • NONE

Property	Type	Required	Editable	Description	Constraint
GroupDeletionProtection	Boolean	No	Yes	Specifies whether to enable deletion protection for the scaling group.	Default value: false. Valid values: • true: enables deletion protection for the scaling group. In this case, you cannot delete the scaling group. • false: disables deletion protection for the scaling group.
DesiredCapacity	Integer	No	Yes	The expected number of ECS instances in the scaling group. The scaling group automatically keeps the number of ECS instances at the expected value.	The number of ECS instances must be greater than the MinSize value and less than the MaxSize value.
InstanceId	String	No	No	The ID of the ECS instance from which the scaling group obtains configuration information to create scaling configurations.	None

Response parameters

Fn::GetAtt

ScalingGroupId: the ID of the scaling group. This ID is a globally unique identifier (GUID) that is generated by the system.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ScalingGroup": {
      "Type": "ALIYUN::ESS::ScalingGroup",
      "Properties": {
        "MaxSize": 1,
        "MinSize": 1,
        # "ScalingGroupName": "HeatCreatedR****",
        # "DefaultCooldown": 500,
        # "RemovalPolicy_1": "",
        # "RemovalPolicy_2": "",
      }
    }
  },
  "Outputs": {
    "ScalingGroup": {
         "Value": {"Fn::GetAtt": ["ScalingGroup", "ScalingGroupId"]}
    }
  }
}
```

5.1.5.2.6. ALIYUN::ESS::ScalingGroupEnable

ALIYUN::ESS::ScalingGroupEnable is used to enable a scaling group.

Syntax

```
{
  "Type": "ALIYUN::ESS::ScalingGroupEnable",
  "Properties": {
    "ScalingConfigurationId": String,
    "ScalingRuleArisExecuteVersion": Integer,
    "ScalingRuleAris": List,
    "ScalingGroupId": String,
    "RemoveInstanceIds": List,
    "InstanceIds": List
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
ScalingGroupId	String	Yes	No	The ID of the scaling group.	None
ScalingConfigurationId	String	No	No	The ID of the scaling configuration to be activated in the scaling group.	None

Property	Type	Required	Editable	Description	Constraint
InstanceIds	List	No	Yes	The IDs of ECS instances to be added to the enabled scaling group.	A maximum of 20 instance IDs can be specified.
ScalingRuleArisExecuteVersion	Integer	No	Yes	The version of the identifier for the scaling rule to be executed. If you change this property, all scaling rules specified by ScalingRuleAris will be executed once.	Minimum value: 0.
ScalingRuleAris	List	No	Yes	The unique identifiers of scaling rules in the scaling group. Invalid unique identifiers are not displayed in the query results and no errors are reported.	A maximum of 10 scaling rule identifiers can be specified.
RemoveInstanceIds	List	No	Yes	The IDs of ECS instances to be deleted.	A maximum of 1,000 instance IDs can be specified.

Response parameters

Fn::GetAtt

- LifecycleState: the status of the scaling group.
- ScalingInstances: the instances that are automatically created in the scaling group.
- ScalingGroupId: the ID of the scaling group.
- ScalingRuleArisExecuteResultInstancesRemoved: the instances that are removed from the scaling group by executing the scaling rules specified by ScalingRuleAris.
- ScalingRuleArisExecuteResultNumberOfAddedInstances: the number of instances that are added to the scaling group by executing the scaling rules specified by ScalingRuleAris.
- ScalingInstanceDetails: the instance scaling details.
- ScalingRuleArisExecuteErrorInfo: the error information about the execution of the scaling rules specified by ScalingRuleAris.
- ScalingRuleArisExecuteResultInstancesAdded: the instances that are added to the scaling group by executing the scaling rules specified by ScalingRuleAris.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ScalingGroupEnable": {
      "Type": "ALIYUN::ESS::ScalingGroupEnable",
      "Properties": {
        "ScalingGroupId": "r0HUqbJ411cc2eQw8bU****",
        "ScalingConfigurationId": "bJlLfdexm77Ldsyptmel****",
        "InstanceIds": "",
      }
    }
  },
  "Outputs": {
    "ScalingGroupEnable": {
      "Value": {"Fn::GetAtt": ["ScalingGroupEnable", "LifecycleState"]}
    }
  }
}
```

5.1.5.2.7. ALIYUN::ESS::ScalingRule

ALIYUN::ESS::ScalingRule is used to create a scaling rule.

Syntax

```
{
  "Type": "ALIYUN::ESS::ScalingRule",
  "Properties": {
    "AdjustmentValue": Integer,
    "Cooldown": Integer,
    "ScalingGroupId": String,
    "AdjustmentType": String,
    "ScalingRuleName": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
AdjustmentValue	Integer	No	Yes	The number of ECS instances to add or release when scaling occurs. The number of ECS instances to be adjusted in a single scaling activity cannot exceed 500.	Valid values in different adjustment modes: • QuantityChangeInCapacity: -500 to 500. • PercentChangeInCapacity: -100 to 10000. • TotalCapacity: 0 to 1000.
Cooldown	Integer	No	Yes	The cooldown period of the scaling rule. Unit: seconds.	Valid values: 0 to 86400. This parameter is empty by default.

Property	Type	Required	Editable	Description	Constraint
ScalingGroupId	String	Yes	No	The ID of the scaling group to which the scaling rule belongs.	None
AdjustmentType	String	Yes	Yes	The adjustment mode of the scaling rule.	Valid values: • QuantityChangeInCapacity: adds or removes a specified number of ECS instances. • PercentChangeInCapacity: adds or removes a specified proportion of ECS instances. • TotalCapacity: adds or removes ECS instances to ensure that the current scaling group has a specified number of ECS instances.
ScalingRuleName	String	No	Yes	The display name of the scaling rule.	The name must be 2 to 40 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit. The name of a scaling rule must be unique within the scaling group that it belongs to. The default value is the ID of the scaling rule.

Response parameters

Fn::GetAtt

- ScalingRuleAri: the unique identifier of the scaling rule.
- ScalingRuleId: the ID of the scaling rule. It is a globally unique identifier (GUID) generated by the system.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ScalingRule": {
      "Type": "ALIYUN::ESS::ScalingRule",
      "Properties": {
        "ScalingRuleName": {
          "Ref": "ScalingRuleName"
        },
        "Cooldown": {
          "Ref": "Cooldown"
        },
        "ScalingGroupId": {
```

```
          "Ref": "ScalingGroupId"
        },
        "AdjustmentType": {
          "Ref": "AdjustmentType"
        },
        "AdjustmentValue": {
          "Ref": "AdjustmentValue"
        }
      }
    }
  },
  "Parameters": {
    "ScalingRuleName": {
      "AllowedPattern": "^[a-zA-Z0-9\\u4e00-\\u9fa5][-_.a-zA-Z0-9\\u4e00-\\u9fa5]{1,63}$",
      "Type": "String",
      "Description": "Name shown for the scaling group, which is a string containing 2 to 40 English or Chinese characters. It must begin with a number, a letter (case-insensitive) or a Chinese character and can contain numbers, \"_\", \"-\" or \". \". The account name in the same scaling group is unique in the same region. If this parameter value is not specified, the default value is ScalingRuleId."
    },
    "Cooldown": {
      "Type": "Number",
      "Description": "Cool-down time of a scaling rule. Value range: [0, 86,400], in seconds. The default value is empty.",
      "MaxValue": 86400,
      "MinValue": 0
    },
    "ScalingGroupId": {
      "Type": "String",
      "Description": "ID of the scaling group of a scaling rule."
    },
    "AdjustmentType": {
      "Type": "String",
      "Description": "Adjustment mode of a scaling rule. Optional values:\n- QuantityChangeInCapacity: It is used to increase or decrease a specified number of ECS instances.\n- PercentChangeInCapacity: It is used to increase or decrease a specified proportion of ECS instances.\n- TotalCapacity: It is used to adjust the quantity of ECS instances in the current scaling group to a specified value.",
      "AllowedValues": [
        "QuantityChangeInCapacity",
        "PercentChangeInCapacity",
        "TotalCapacity"
      ]
    },
    "AdjustmentValue": {
      "Type": "Number",
      "Description": "Adjusted value of a scaling rule. Value range:\n- QuantityChangeInCapacity: [-500, 500]\n- PercentChangeInCapacity: [-100, 10000]\n- TotalCapacity: [0, 1000]",
      "MaxValue": 10000,
      "MinValue": -500
    }
  },
  "Outputs": {
    "ScalingRuleAri": {
      "Description": "Unique identifier of a scaling rule.",
      "Value": {
        "Fn::GetAtt": [
          "ScalingRule",
          "ScalingRuleAri"
        ]
```

```
      }
    },
    "ScalingRuleId": {
      "Description": "ID of a scaling rule, generated by the system and globally unique.",
      "Value": {
        "Fn::GetAtt": [
          "ScalingRule",
          "ScalingRuleId"
        ]
      }
    }
  }
}
```

5.1.5.2.8. ALIYUN::ESS::ScheduledTask

ALIYUN::ESS::ScheduledTask is used to create a scheduled task based on input parameters.

Syntax

```
{
  "Type": "ALIYUN::ESS::ScheduledTask",
  "Properties": {
    "TaskEnabled": Boolean,
    "Description": String,
    "ScheduledTaskName": String,
    "LaunchExpirationTime": Integer,
    "LaunchTime": String,
    "RecurrenceEndTime": String,
    "RecurrenceType": String,
    "RecurrenceValue": String,
    "ScheduledAction": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
TaskEnabled	Boolean	No	Yes	Specifies whether to start the scheduled task. • true: starts the scheduled task. • false: stops the scheduled task. Default value: true.	None
Description	String	No	Yes	The description of the scheduled task.	The description must be 2 to 200 characters in length.

Property	Type	Required	Editable	Description	Constraint
ScheduledTaskName	String	No	Yes	The display name of the scheduled task.	The name must be 2 to 40 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit. This parameter must be unique in a region and under an Apsara Stack tenant account. The default value is the ID of the scheduled scaling task.
LaunchExpirationTime	Integer	No	Yes	The time period during which a failed scheduled task is retried. Unit: seconds. Default value: 600.	Valid values: 0 to 21600.
LaunchTime	String	Yes	Yes	The time at which the scheduled task is triggered. Specify the time in the ISO 8601 standard in the YYYY-MM-DDThh:mmZ format. The time must be in UTC. If the RecurrenceType parameter is specified, the task is executed each day at the time specified by LaunchTime. If the RecurrenceType parameter is not specified, the task is only executed once at the date and time specified by LaunchTime. You cannot enter a point in time later than 90 days from the date of scheduled task creation or modification.	None

Property	Type	Required	Editable	Description	Constraint
RecurrenceEndTime	String	No	Yes	The end time after which the scheduled task will not be repeated. Specify the time in the ISO 8601 standard in the YYYY-MM-DDThh:mmZ format. The time must be in UTC. You cannot enter a point in time later than 90 days from the date of scheduled task creation or modification. If you set RecurrenceEndTime, you must also set both RecurrenceType and RecurrenceValue.	None
RecurrenceType	String	No	Yes	The interval that the scheduled task is repeated at.	Valid values: • Daily: The scheduled task is executed once every specified number of days. • Weekly: The scheduled task is executed on each specified day of a week. • Monthly: The scheduled task is executed on each specified day of a month. • Cron: The scheduled task is executed based on the specified Cron expression. If you set RecurrenceType, you must also set both RecurrenceEndTime and RecurrenceValue.

Property	Type	Required	Editable	Description	Constraint
RecurrenceValue	String	No	Yes	Specifies how often the scheduled task recurs.	• Daily: indicates the interval of days that the scheduled task is repeated on. You can enter a single value ranging from 1 to 31. • Weekly: indicates which days of the week that the scheduled task is repeated on. You can enter multiple values separated by commas (,). The values 0 to 6 correspond to the days of the week in sequence from Sunday to Saturday. • Monthly: indicates which days of the month that the scheduled task is repeated on. You can enter two values ranging from 1 to 31. The format is A-B. B must be greater than or equal to A. • Cron: indicates a user-defined Cron expression that the scheduled task is repeated on. A Cron expression is written in UTC time and consists of five fields: minute, hour, day of month (date), month, and day of week. The expression can contain wildcard characters including commas (,), question marks (?), hyphens (-), asterisks (*), number signs (#), forward slashes (/), and the L and W characters. If you set RecurrenceValue, you must also set both RecurrenceEndTime and RecurrenceType.
ScheduledAction	String	Yes	Yes	The operations to be performed when the scheduled task is triggered. When you set this parameter, you must also enter the unique identifier of the scaling rule.	The parameter value can be up to 200 characters in length.

Response parameters

Fn::GetAtt

ScheduledTaskId: the ID of the scheduled task. This ID is a globally unique identifier (GUID) generated by the system.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ScheduledTask": {
      "Type": "ALIYUN::ESS::ScheduledTask",
      "Properties": {
        "TaskEnabled": "true",
        "Description": "scheduledtask",
        "ScheduledTaskName": "task1",
        "LaunchTime": "2014-08-17T16:52Z",
        "RecurrenceEndTime": "2014-08-17T16:55Z",
        "RecurrenceType": "Daily",
        "RecurrenceValue": "1",
        "ScheduledAction": "ari:acs:ess:cn-qingdao:1344371:scalingRule/cCBpdYdQuBe2cUxOdu6piOk"
      }
    }
  },
  "Outputs": {
    "ScheduledTaskId": {
      "Value": {
        "FN::GetAtt": [
          "ScheduledTask",
          "ScheduledTaskId"
        ]
      }
    }
  }
}
```

5.1.5.3. OSS

5.1.5.3.1. ALIYUN::OSS::Bucket

ALIYUN::OSS::Bucket is used to create an OSS bucket.

Syntax

```
{
  "Type": "ALIYUN::OSS::Bucket",
  "Properties": {
    "AccessControl": String,
    "RefererConfiguration": Map,
    "ServerSideEncryptionConfiguration": Map,
    "CORSConfiguration": Map,
    "Tags": Map,
    "LoggingConfiguration": Map,
    "LifecycleConfiguration": Map,
    "StorageClass": String,
    "DeletionForce": Boolean,
    "WebsiteConfiguration": Map,
    "Policy": Map,
    "BucketName": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
BucketName	String	Yes	No	The name of the bucket.	• The name must be 3 to 63 characters in length and can contain lowercase letters, digits, and hyphens (-). • It must start and end with a lowercase letter or digit.
AccessControl	String	No	No	The access control policy.	Valid values: private, public-read, and public-read-write.
CORSConfiguration	Map	No	No	The configuration of cross-origin resource sharing for objects in the bucket.	None
LifecycleConfiguration	Map	No	No	The lifecycle configuration for objects in the bucket.	None
LoggingConfiguration	Map	No	No	The logging configuration.	None
RefererConfiguration	Map	No	No	The hotlinking protection configuration.	None

Property	Type	Required	Editable	Description	Constraint
DeletionForce	Boolean	No	No	Specifies whether to forcibly delete objects from an OSS bucket	Valid values: • true • false Default value: false.
WebsiteConfiguration	Map	No	No	The information used to configure the bucket as a static website.	None
ServerSideEncryptionConfiguration	Map	No	No	The server-side encryption rules.	None
Tags	Map	No	No	The tags of the bucket. Tags exist as key-value pairs.	• A maximum of 20 tags can be specified. • A tag key must be 1 to 64 bytes in length and cannot start with `http://`, `https://`, or `Aliyun`. • A tag value can be up to 128 bytes in length and must be encoded in UTF-8.
StorageClass	String	No	No	The type of the bucket.	Valid values: Standard, IA, and Archive.
Policy	Map	No	No	The bucket policy configuration.	None

CORSConfiguration syntax

```
"CORSConfiguration": {
  "CORSRule": [
    {
      "AllowedHeader": String,
      "AllowedMethod": List,
      "AllowedOrigin": List,
      "ExposeHeader": List,
      "MaxAgeSeconds": Integer
    }
  ]
}
```

CORSConfiguration properties

Property	Type	Required	Editable	Description	Constraint

Property	Type	Required	Editable	Description	Constraint
CORSRule	List	No	No	The rules that define cross-origin resource sharing of objects in the bucket.	None
AllowedHeader	String	No	No	The allowed cross-origin request headers.	Valid values: *, Cache-Control, Content-Language, Content-Type, Expires, Last-Modified, and Pragma.
AllowedMethod	List	No	No	The allowed cross-origin request methods.	Valid values: *, GET, PUT, POST, DELETE, and HEAD.
AllowedOrigin	List	No	No	The origins from which cross-origin requests are allowed.	None
ExposeHeader	List	No	No	The response headers for allowed access requests from applications.	Asterisks (*) cannot be used as wildcard characters.
MaxAgeSeconds	Integer	No	No	The period of time that the browser can cache the response of a preflight (OPTIONS) request to a specific resource.	None

LifecycleConfiguration syntax

```
"LifecycleConfiguration": {
  "Rule": [
    {
      "ID": String,
      "Prefix": String,
      "Status": String,
      "Expiration": Map,
      "AbortMultipartUpload": Map
    }
  ]
}
```

LifecycleConfiguration properties

Property	Type	Required	Editable	Description	Constraint
Rule	List	No	No	The lifecycle rule.	None
ID	String	No	No	The unique ID of the rule.	The ID can be up to 255 characters in length. When this parameter is empty or not specified, OSS generates a unique rule ID.
Prefix	String	No	No	The prefix to which the rule applies.	The rule takes effect only on objects that have a matching prefix.
Status	String	No	No	Specifies whether to enable or disable the rule.	Valid values: Enable and Disable.
Expiration	Map	No	No	The expiration attributes of the rule for the specified object.	None

Property	Type	Required	Editable	Description	Constraint
AbortMultipartUpload	Map	No	No	The expiration attributes of the multipart upload tasks that are not complete.	None

Expiration syntax

```
"Expiration":{
  "Days": Number,
  "CreatedBeforeDate": String
}
```

Expiration properties

Property	Type	Required	Editable	Description	Constraint
Days	Number	No	No	The number of days since the object was last modified after which the rule will take effect.	When the number of days since the object was last modified exceeds the specified number of days, the object is deleted. If you set the Days parameter to 30, objects that were last modified on January 1, 2016 are deleted by the backend application on January 31, 2016.
CreatedBeforeDate	String	No	No	The date before which the rule takes effect.	Specify the time in the ISO 8601 standard. The time must be UTC 00:00. Example: 2002-10-11T00:00:00.000Z.

AbortMultipartUpload syntax

```
"AbortMultipartUpload": {
  "CreatedBeforeDate": String,
  "Days": Number
}
```

AbortMultipartUpload properties

Property	Type	Required	Editable	Description	Constraint

Property	Type	Required	Editable	Description	Constraint
Days	Number	No	No	The number of days since the object was last modified after which the rule will take effect.	When the number of days since the object was last modified exceeds the specified number of days, the object is deleted. If you set the Days parameter to 30, objects that were last modified on January 1, 2016 are deleted by the backend application on January 31, 2016.
CreatedBeforeDate	String	No	No	The date before which the rule takes effect.	Specify the time in the ISO 8601 standard. The time must be UTC 00:00. Example: 2002-10-11T00:00:00.000Z.

LoggingConfiguration syntax

```
"LoggingConfiguration": {
  "TargetBucket": String,
  "TargetPrefix": String
}
```

LoggingConfiguration properties

Property	Type	Required	Editable	Description	Constraint
TargetBucket	String	No	No	The storage space for storing access logs.	None
TargetPrefix	String	No	No	The prefix of the names of saved access log files.	None

WebsiteConfiguration syntax

```
"WebsiteConfiguration":{
  "IndexDocument": String,
  "ErrorDocument": String
}
```

WebsiteConfiguration properties

Property	Type	Required	Editable	Description	Constraint
IndexDocument	String	No	No	The default homepage for a static website.	None

Property	Type	Required	Editable	Description	Constraint
ErrorDocument	String	No	No	The default error page for a static website.	None

RefererConfiguration syntax

```
"RefererConfiguration":{
  "AllowEmptyReferer": String,
  "RefererList": List
}
```

RefererConfiguration properties

Property	Type	Required	Editable	Description	Constraint
AllowEmptyReferer	String	No	No	Specifies whether the Referer field can be left empty in an access request.	None
RefererList	List	No	No	The referer whitelist. OSS allows requests whose Referer field values are in the referer whitelist.	None

ServerSideEncryptionConfiguration syntax

```
"ServerSideEncryptionConfiguration":{
  "KMSMasterKeyID": String,
  "SSEAlgorithm": String
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
KMSMasterKeyID	String	No	No	The ID of the customer master key.	The key ID is required only when the SSEAlgorithm value is KMS and the specified key is used for encryption.
SSEAlgorithm	String	Yes	No	The default server-side encryption method.	Valid values: KMS and AES256.

Response parameters

Fn::GetAtt

- Name: the bucket name, which must be globally unique.
- DomainName: the public domain name of the specified bucket.
- InternalDomainName: the internal domain name of the specified bucket.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Bucket": {
      "Type": "ALIYUN::OSS::Bucket",
      "Properties": {
        "AccessControl": "private",
        "BucketName": "rostest",
        "WebsiteConfiguration":{
            "IndexDocument": "index1.html",
            "ErrorDocument": "error404.html"
        },
        "LoggingConfiguration": {
            "TargetBucket": "cos-mirror",
            "TargetPrefix": "test404"
        },
        "CORSConfiguration": {
            "CORSRule": [{
                "AllowedHeader": ["*"],
                "AllowedMethod": ["GET", "PUT"],
                "AllowedOrigin": ["*"],
                "ExposeHeader": ["Date"],
                "MaxAgeSeconds": 3600
            }]
        },
        "LifecycleConfiguration": {
            "Rule": [{
                "ID": "deleteRule",
                "Prefix": "test/",
                "Status": "Enabled",
                "Expiration":{
                    "Days": 2
                },
                "AbortMultipartUpload":{
                    "CreatedBeforeDate": "2014-10-11T00:00:00.000Z"
                }
            }]
        },
        "RefererConfiguration": {
            "AllowEmptyReferer": true,
            "RefererList": ["http://www.aliyun.com", "https://www.?.aliyuncs.com"]
        }
      }
    }
  },
  "Outputs": {
    "Name": {
         "Value": {"Fn::GetAtt": ["Bucket","Name"]}
    },
    "DomainName": {
         "Value": {"Fn::GetAtt": ["Bucket","DomainName"]}
    }
  }
}
```

5.1.5.4. RDS

5.1.5.4.1. ALIYUN::RDS::Account

ALIYUN::RDS::Account is used to create a database management Account.

Statement

```
{
  "Type": "ALIYUN::RDS::Account",
  "Properties": {
    "AccountDescription": String,
    "DBInstanceId": String,
    "AccountPassword": String,
    "AccountType": String,
    "AccountName": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
AccountDescription	String	Yes	True	The description of the account.	The name must be 2 to 256 characters in length. It can contain digits, letters, underscores (_), and hyphens (-); but must start with a letter.
DBInstanceId	String	No	No	The ID of the RDS instance.	None
AccountPassword	String	No	No	The password of the database account.	The password must be 8 to 32 characters in length.
AccountType	String	Yes	Released	The type of the database account.	Valid values: • Normal: indicates a standard account. • Super: indicates a privileged account. Default value: Normal.

Parameter	Type	Required	Editable	Description	Constraint
AccountName	String	No	No	The name of the database account.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

AccountName: the name of the database account.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Account": {
      "Type": "ALIYUN::RDS::Account",
      "Properties": {
        "AccountDescription": {
          "Ref": "AccountDescription"
        },
        "DBInstanceId": [
          "Ref": "DBInstanceId"
        },
        "AccountPassword": {
          "Ref": "AccountPassword"
        },
        "AccountType": {
          "Ref": "AccountType"
        },
        "AccountName": {
          "Ref": "AccountName"
        }
      }
    }
  },
  "Parameters": {
    "AccountDescription": {
      "Type": "String",
      "Description": "Account remarks.\nIt cannot begin with http:// or https://.\nIt must start with a Chinese character or English letter.\nIt can include Chinese and English characters/letters, underscores (_), hyphens (-), and digits.\nThe length may be 2-256 characters."
    },
    "DBInstanceId": [
      "Type": "String",
      "Description": "RDS instance ID."
    },
    "AccountPassword": {
      "MinLength": 8,
      "Type": "String",
      "Description": "The account password for the database instance. It may consist of letters, digi
```

```
ts, or underlines, with a length of 8 to 32 characters.",
      "MaxLength": 32
    },
    "AccountType": {
      "Default": "Normal",
      "Type": "String",
      "Description": "Privilege type of account.\nNormal: Common privilege.\nSuper: High privilege. A
nd the default value is Normal.\nThis parameter is valid for MySQL 5.5/5.6 only.\nMySQL 5.7, SQL Serv
er 2012/2016, PostgreSQL, and PPAS each can have only one initial account. Other accounts are created
by the initial account that has logged on to the database.",
      "AllowedValues": ["Normal", "Super"]
    },
    "AccountName": {
      "Type": "String",
      "Description": "Account name, which must be unique and meet the following requirements:\nStart
with a letter;\nConsist of lower-case letters, digits, and underscores (_);\nContain no more than 16
characters.\nFor other invalid characters, see Forbidden keywords table."
    }
  },
  "Outputs": {
    "AccountName": {
      "Description": "Account name",
      "Value": {
        "Fn::GetAtt": ["Account", "AccountName"]
      }
    }
  }
}
```

5.1.5.4.2. ALIYUN::RDS::AccountPrivilege

ALIYUN::RDS::AccountPrivilege is used to grant database access permissions to accounts.

Statement

```
{
  "Type": "ALIYUN::RDS::AccountPrivilege",
  "Properties": {
    "AccountPrivilege": String,
    "DBInstanceId": String,
    "DBName": String,
    "AccountName": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint

Parameter	Type	Required	Editable	Description	Constraint
AccountPrivilege	String	No	Yes	The permissions of the database account.	Valid values: • ReadWrite: has read and write permissions on the database. • ReadOnly: The account has read-only permission on the database. • DDLOnly: The account can run only data definition language (DDL) commands in the database. This is applicable to MySQL and MariaDB. • DMLOnly: The account can run only data manipulation language (DML) commands in the database. This is applicable to MySQL and MariaDB. • DBOwner: The account has full permissions on the database. This is applicable to SQL Server.
DBInstanceId	String	No	No	The ID of the RDS instance.	None
DBName	String	No	No	The name of the database.	None

Parameter	Type	Required	Editable	Description	Constraint
AccountName	String	No	No	The name of the account.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

None

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "AccountPrivilege": {
      "Type": "ALIYUN::RDS::AccountPrivilege",
      "Properties": {
        "AccountPrivilege": {
          "Ref": "AccountPrivilege"
        },
        "DBInstanceId": [
          "Ref": "DBInstanceId"
        },
        "DBName": {
          "Ref": "DBName"
        },
        "AccountName": {
          "Ref": "AccountName"
        }
      }
    }
  },
  "Parameters": {
    "AccountPrivilege": {
      "Type": "String",
      "Description": "RDS account privilege",
      "AllowedValues": ["ReadOnly", "ReadWrite", "DDLOnly", "DMLOnly", "DBOwner"]
    },
    "DBInstanceId": [
      "Type": "String",
      "Description": "RDS instance ID."
    },
    "DBName": {
      "Type": "String",
      "Description": "RDS database name"
    },
    "AccountName": {
      "Type": "String",
      "Description": "RDS account name."
    }
  },
  "Outputs": {}
}
```

5.1.5.4.3. ALIYUN::RDS::DBInstance

ALIYUN::RDS::DBInstance is used to create an ApsaraDB RDS instance.

Syntax

```
{
  "Type": "ALIYUN::RDS::DBInstance",
  "Properties": {
    "Engine": String,
    "MultiAZ": Boolean,
    "VpcId": String,
    "DBMappings": List,
    "DBInstanceDescription": String,
    "ConnectionMode": String,
    "MasterUsername": String,
    "MasterUserPassword": String,
    "ZoneId": String,
    "DBInstanceNetType": String,
    "DBInstanceStorage": Integer,
    "VSwitchId": String,
    "AllocatePublicConnection": Boolean,
    "EngineVersion": String,
    "PreferredBackupTime": String,
    "DBInstanceClass": String,
    "SecurityIPList": String,
    "BackupRetentionPeriod": Integer,
    "PrivateIpAddress": String,
    "PreferredBackupPeriod": List,
    "PeriodType": String,
    "PayType": String,
    "Period": Integer,
    "ResourceGroupId": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	No	No	The ID of the resource group.	None
Engine	String	Yes	No	The database engine that the instance runs.	Valid values: • MySQL • SQLServer • PostgreSQL • PPAS

Property	Type	Required	Editable	Description	Constraint
DBInstanceStorage	Integer	Yes	Yes	The storage capacity of the instance.	• Valid values when Engine is set to MySQL: 5 to 1000. • Valid values when Engine is set to SQLServer: 10 to 1000. • Valid values when Engine is set to PostgreSQL: 5 to 2000. • Valid values when Engine is set to PPAS: 5 to 2000. Unit: GB. **Note** This value must be in 5 GB increments.
EngineVersion	String	Yes	No	The version of the database engine.	• Valid values when Engine is set to MySQL: 5.5, 5.6, 5.7, and 8.0. • Set the value to 2008r2 when Engine is set to SQLServer. • Set the value to 9.4 when Engine is set to PostgreSQL. • Set the value to 9.3 when Engine is set to PPAS.
DBInstanceClass	String	Yes	Yes	The instance type.	Valid values: • rds.mys2.large • rds.mss1.large • rds.pg.s1.small
SecurityIPList	String	Yes	Yes	The whitelist of IP addresses that are allowed to access all databases in the instance.	• Separate multiple IP addresses with commas (,). Each IP address in the whitelist must be unique. A maximum of 1,000 IP addresses can be specified. • The 0.0.0.0/0 format is supported. You can specify IP addresses in the 10.23.XX.XX format and CIDR blocks in the 10.23.XX.XX/24 format. In 10.23.XX.XX/24, /24 indicates the length of the prefix in the CIDR block, and the prefix length can range from 1 to 32. 0.0.0.0/0 indicates that no access restriction is applied.
MultiAZ	Boolean	No	No	Specifies whether the instance can be deployed across multiple zones.	None
VpcId	String	No	No	The ID of the VPC.	None

Property	Type	Required	Editable	Description	Constraint
DBMappings	List	No	No	The list of one or more databases to be created in the instance.	None
DBInstanceDescription	String	No	No	The description of the instance.	• The description must be 2 to 256 characters in length and can contain letters, digits, underscores (_), and hyphens (-). • It must start with a letter and cannot start with `http://` or `https://`.
ConnectionMode	String	No	No	The connection mode of the instance.	Valid values: • Performance: standard connection mode • Safty: safe connection mode If you do not specify this parameter, the system assigns a connection mode.
MasterUsername	String	No	No	The name of the database account.	The name must be unique. The name can be up to 16 characters in length and can contain letters, digits, and underscores (_).
MasterUserPassword	String	No	No	The password of the database account.	The password must be 6 to 32 characters in length and can contain letters, digits, and underscores (_).
ZoneId	String	No	No	The zone ID of the instance.	None
DBInstanceNetType	String	No	No	The network type of the instance.	Default value: Intranet. Valid values: • Internet • Intranet
VSwitchId	String	No	No	The ID of the vSwitch in the specified VPC.	None
AllocatePublicConnection	Boolean	No	No	Specifies whether to apply for a public endpoint for the instance.	None

Property	Type	Required	Editable	Description	Constraint
PreferredBackupTime	String	No	No	The backup window.	• Specify the window in the `HH:mmZ-HH:mmZ` format. • Valid values: 00:00Z-01:00Z, 01:00Z-02:00Z, 02:00Z-03:00Z, 03:00Z-04:00Z, 04:00Z-05:00Z, 05:00Z-06:00Z, 06:00Z-07:00Z, 07:00Z-08:00Z, 08:00Z-09:00Z, 09:00Z-10:00Z, 10:00Z-11:00Z, 11:00Z-12:00Z, 12:00Z-13:00Z, 13:00Z-14:00Z, 14:00Z-15:00Z, 15:00Z-16:00Z, 16:00Z-17:00Z, 17:00Z-18:00Z, 18:00Z-19:00Z, 19:00Z-20:00Z, 20:00Z-21:00Z, 21:00Z-22:00Z, 22:00Z-23:00Z, and 23:00Z-24:00Z.
BackupRetentionPeriod	Number	No	No	The number of days for which backup files can be retained.	Valid values: 7 to 30. Unit: days. Default value: 7.
PrivateIpAddress	String	No	No	The private IP address within the CIDR block of the vSwitch.	If you do not specify this parameter, the system allocates a private IP address.
PreferredBackupPeriod	List	No	No	The backup cycle.	Valid values: • Monday • Tuesday • Wednesday • Thursday • Friday • Saturday • Sunday
MasterUserType	String	No	No	The type of the database account.	Default value: Normal. Valid values: • Normal • Super
Tags	Map	No	Yes	The list of one or more tags. Each tag consists of a tag key and a tag value.	• The tag key is required and the tag value is optional. • Format example: `{"key1":"value1","key2":""}`.
PeriodType	String	No	No	The unit of the subscription period.	Default value: Month. Valid values: • Month • Year
PayType	String	No	No	The billing method of the instance.	Valid values: • PostPaid: pay-as-you-go • PrePaid: subscription

Property	Type	Required	Editable	Description	Constraint
Period	Integer	No	No	The subscription period of the instance.	• Valid values when PeriodType is set to Year: 1, 2, and 3. • Valid values when PeriodType is set to Month: 1, 2, 3, 4, 5, 6, 7, 8, and 9.

DBMappings syntax

```
"DBMappings": [
  {
    "DBDescription": String,
    "CharacterSetName": String,
    "DBName": String
  }
]
```

DBMappings properties

Property	Type	Required	Editable	Description	Constraint

CharacterSetName	String	Yes	No	The character set.	• Valid values when Engine is set to MySQL:
					  ◦ utf8
					  ◦ gbk
					  ◦ latin1
					  ◦ utf8mb4 (applicable to versions 5.5 and 5.6)
					• Valid values when Engine is set to SQLServer:
					  ◦ Chinese_PRC_CI_AS
					  ◦ Chinese_PRC_CS_AS
					  ◦ SQL_Latin1_General_CP1_CI_AS
					  ◦ SQL_Latin1_General_CP1_CS_AS
					  ◦ Chinese_PRC_BIN

DBName	String	Yes	No	The name of the database.	The name must be unique. The name can be up to 64 characters in length and can contain letters, digits, and underscores (_). It must start with a letter.

Property	Type	Required	Editable	Description	Constraint
DBDescription	String	No	No	The description of the database.	• The description must be 2 to 256 characters in length and can contain letters, digits, underscores (_), and hyphens (-). • It must start with a letter and cannot start with `http://` or `https://`.

Response parameters

Fn::GetAtt

- DBInstanceId: the ID of the instance.
- InnerPort: the internal port of the instance.
- InnerIPAddress: the internal IP address of the instance.
- InnerConnectionString: the internal endpoint of the instance.
- PublicPort: the public port of the instance.
- PublicConnectionString: the public endpoint of the instance.
- PublicIPAddress: the public IP address of the instance.

Examples

The following example demonstrates how to create an ApsaraDB RDS instance in the classic network:

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Database": {
      "Type": "ALIYUN::RDS::DBInstance",
      "Properties": {
        "Engine": "MySQL",
        "EngineVersion": "5.6",
        "DBInstanceClass":"rds.mysql.t1.small",
        "DBInstanceStorage": 10,
        "DBInstanceNetType": "Internet",
        "SecurityIPList": "0.0.0.0/0",
        "MasterUsername": "A****",
        "DBMappings":[{
            "DBName": "hope",
            "CharacterSetName": "utf8"
        }]
      }
    }
  },
  "Outputs": {
    "DBInstanceId": {
      "Value": {"get_attr": ["DBInstanceId"]}
    },
    "PublicConnectionString": {
      "Value": {"get_attr": ["ConnectionString"]}
    },
    "PublicPort": {
      "Value": {"get_attr": ["Port"]}
    }
  }
}
```

The following example demonstrates how to create an ApsaraDB RDS instance in a VPC:

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Database": {
      "Type": "ALIYUN::RDS::DBInstance",
      "Properties": {
        "Engine":"MySQL",
        "EngineVersion":"5.6",
        "DBInstanceClass":"rds.mys2.small",
        "DBInstanceStorage":"10",
        "DBInstanceNetType":"Intranet",
        "SecurityIPList": "0.0.0.0/0",
        "VSwitchId": "ttt",
        "VpcId": "myvp****"
      }
    }
  },
  "Outputs": {
    "DBInstanceId": {
      "Value": {"get_attr": ["DBInstanceId"]}
    },
    "InnerConnectionString": {
      "Value": {"get_attr": ["ConnectionString"]}
    },
    "InnerPort": {
      "Value": {"get_attr": ["Port"]}
    }
  }
}
```

5.1.5.4.4. ALIYUN::RDS::DBInstanceParameterGroup

ALIYUN::RDS::DBInstanceParameterGroup is used to modify parameters of an ApsaraDB RDS instance.

Syntax

```
{
  "Type": "ALIYUN::RDS::DBInstanceParameterGroup",
  "Properties": {
    "Forcerestart": String,
    "DBInstanceId": String,
    "Parameters": List
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
DBInstanceId	String	Yes	No	The ID of the ApsaraDB RDS instance.	None

Property	Type	Required	Editable	Description	Constraint
Parameters	List	Yes	No	The list of one or more parameters of the instance.	The parameters and their values must be arranged in the JSON format. The parameter values must be of the string type. Example: {"auto_increment_increment":"1","character_set_client":"utf8"}.
Forcerestart	String	No	No	Specifies whether to forcibly restart the instance.	Default value: false. Valid values: • true: The system forcibly restarts the instance. • false: The system does not forcibly restart the instance.

Response parameters

Fn::GetAtt

None

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Database": {
      "Type": "ALIYUN::RDS::DBInstance",
      "Properties": {
        "Engine": "MySQL",
        "EngineVersion": "5.6",
        "DBInstanceClass": "rds.mys2.small",
        "DBInstanceStorage": "10",
        "DBInstanceNetType": "Intranet",
        "SecurityIPList": "0.0.0.0/0"
      }
    },
    "DatabaseConfig": {
      "Type": "ALIYUN::RDS::DBInstanceParameterGroup",
      "Properties": {
        "DBInstanceId": {
          "Ref": "Database"
        },
        "Parameters": [
          {
            "Key": "auto_increment_increment",
            "Value": "xxx"
          }
        ]
      }
    }
  },
  "Outputs": {
    "DBInstanceId": {
      "Value": {
        "Fn::GetAtt": [
          "Database",
          "DBInstanceId"
        ]
      }
    }
  }
}
```

5.1.5.4.5. ALIYUN::RDS::DBInstanceSecurityIps

ALIYUN::RDS::DBInstanceSecurityIps is used to modify the instance whitelist.

Statement

```
{
  "Type": "ALIYUN::RDS::DBInstanceSecurityIps",
  "Properties": {
    "DBInstanceId": String,
    "DBInstanceIPArrayName": String,
    "DBInstanceIPArrayAttribute": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
DBInstanceId	String	No	No	The ID of the RDS instance.	None
DBInstanceIPArrayAttribute	String	No	Yes	The attribute of the IP address whitelist.	The console does not display groups labeled with hidden.
DBInstanceIPArrayName	String	Yes	Released	The name of the IP address whitelist.	The name can contain only lowercase letters and underscores (_). Default value: Default.

Response parameters

Fn::GetAtt

SecurityIps: the IP address whitelist after the modification.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "DBInstanceSecurityIps": {
      "Type": "ALIYUN::RDS::DBInstanceSecurityIps",
      "Properties": {
        "DBInstanceIPArrayName": {
          "Ref": "DBInstanceIPArrayName"
        },
        "DBInstanceId": [
          "Ref": "DBInstanceId"
        },
        "DBInstanceIPArrayAttribute": {
          "Ref": "DBInstanceIPArrayAttribute"
        }
      }
    }
  },
  "Parameters": {
    "DBInstanceIPArrayName": {
      "Type": "String",
      "Description": "Group name of the security ips, only support lower characters and '_'. Advice use a new group name avoid effect your database system. If the properties is not specified, it will set to default group, please be careful."
    },
    "DBInstanceId": [
      "Type": "String",
      "Description": "Database instance id to update security ips."
    },
    "DBInstanceIPArrayAttribute": {
      "Type": "String",
      "Description": "Security ips to add or remove."
    }
  },
  "Outputs": {
    "SecurityIps": {
      "Description": "The security ips of selected database instance.",
      "Value": {
        "Fn::GetAtt": [
          "DBInstanceSecurityIps",
          "SecurityIps"
        ]
      }
    }
  }
}
```

5.1.5.4.6. ALIYUN::RDS::PrepayDBInstance

ALIYUN::RDS::PrepayDBInstance is used to create a subscription ApsaraDB RDS instance.

Syntax

```
{
  "Type": "ALIYUN::RDS::PrepayDBInstance",
  "Properties": {
    "DBMappings": List,
    "CouponCode": String,
    "MasterUsername": String,
    "PeriodType": String,
    "PayType": String,
    "DBInstanceNetType": String,
    "MasterUserType": String,
    "AutoRenew": Boolean,
    "PreferredBackupTime": String,
    "PrivateIpAddress": String,
    "Engine": String,
    "MultiAZ": Boolean,
    "VpcId": String,
    "ConnectionMode": String,
    "ResourceGroupId": String,
    "VSwitchId": String,
    "BackupRetentionPeriod": Number,
    "Quantity": Number,
    "CommodityCode": String,
    "ZoneId": String,
    "AutoPay": Boolean,
    "EngineVersion": String,
    "DBInstanceClass": String,
    "PreferredBackupPeriod": List,
    "DBInstanceStorage": Integer,
    "DBInstanceDescription": String,
    "Tags": Map,
    "Period": Number,
    "MasterUserPassword": String,
    "AllocatePublicConnection": Boolean
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	No	No	The ID of the resource group.	None
DBMappings	List	No	No	The list of one or more databases to be created in the instance.	None
CouponCode	String	No	No	None	None
MasterUsername	String	No	No	The name of the database account.	The name must be unique. The name can be up to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter.

Property	Type	Required	Editable	Description	Constraint
PeriodType	String	Yes	No	The unit of the subscription period.	Default value: Month. Valid values: • Year • Month
DBInstanceNetType	String	No	No	The network type of the instance.	Default value: Intranet. Valid values: • Internet • Intranet
MasterUserType	String	No	No	The type of the database account.	Valid values: • Normal • Master
PreferredBackupTime	String	No	No	The backup window.	Specify the window in the HH:mmZ-HH:mmZ format. Value values: 00:00Z-01:00Z, 01:00Z-02:00Z, 02:00Z-03:00Z, 03:00Z-04:00Z, 04:00Z-05:00Z, 05:00Z-06:00Z, 06:00Z-07:00Z, 07:00Z-08:00Z, 08:00Z-09:00Z, 09:00Z-10:00Z, 10:00Z-11:00Z, 11:00Z-12:00Z, 12:00Z-13:00Z, 13:00Z-14:00Z, 14:00Z-15:00Z, 15:00Z-16:00Z, 16:00Z-17:00Z, 17:00Z-18:00Z, 18:00Z-19:00Z, 19:00Z-20:00Z, 20:00Z-21:00Z, 21:00Z-22:00Z, 22:00Z-23:00Z, and 23:00Z-24:00Z.
PrivateIpAddress	String	No	No	The private IP address with the CIDR block of the specified vSwitch.	If you do not specify this parameter, the system allocates a private IP address.
Engine	String	Yes	No	The database engine that the instance runs.	Valid values: • MySQL • SQLServer • PostgreSQL • PPAS

Property	Type	Required	Editable	Description	Constraint
MultiAZ	Boolean	No	No	Specifies whether the instance can be deployed across multiple zones.	None
VpcId	String	No	No	The ID of the VPC.	None
ConnectionMode	String	No	No	The connection mode of the instance.	Default value: Safty. Valid values: • Performance: the standard mode. • Safty: the database proxy mode. If you do not specify this parameter, the system assigns a connection mode.
AutoRenew	Boolean	No	No	Specifies whether to enable automatic renewal for the instance.	Valid values: • True • False
VSwitchId	String	No	No	The ID of the vSwitch in the specified VPC.	None
BackupRetentionPeriod	Number	No	No	The number of days for which backup files can be retained.	None
Quantity	Number	No	No	The number of instances to be created.	Valid values: 1 to 99. Default value: 1.
CommodityCode	String	Yes	No	The commodity code.	Valid values: • rds • bards • rords
ZoneId	String	No	No	The zone ID of the instance.	None

Property	Type	Required	Editable	Description	Constraint
EngineVersion	String	Yes	No	The version of the database engine.	• Valid values when Engine is set to MySQL: 5.5 and 5.6. • Set the value to 2008r2 when Engine is set to SQLServer. • Set the value to 9.4 when Engine is set to PostgreSQL. • Set the value to 9.3 when Engine is set to PPAS.
DBInstanceClass	String	Yes	Yes	The instance type.	Examples: rds.mys2.large, rds.mss1.large, and rds.pg.s1.small.
PreferredBackupPeriod	List	No	No	The backup cycle.	Valid values: • Monday • Tuesday • Wednesday • Thursday • Friday • Saturday • Sunday
DBInstanceStorage	Integer	Yes	Yes	The storage capacity of the instance.	• Valid values when Engine is set to MySQL: 5 to 1000. • Valid values when Engine is set to SQLServer: 10 to 1000. • Valid values when Engine is set to PostgreSQL or PPAS: 5 to 2000. Unit: GB. This value must be in 5 GB increments.
DBInstanceDescription	String	No	No	The description of the instance.	The description must be 2 to 256 characters in length and can contain letters, digits, underscores (_), and hyphens (-). It must start with a letter and cannot start with `http://` or `https://`.
Tags	map	No	Yes	The tags of the instance.	None

Property	Type	Required	Editable	Description	Constraint
Period	Number	Yes	No	The subscription period of the instance.	• Valid values when PeriodType is set to Month: 1, 2, 3, 4, 5, 6, 7, 8, and 9. • Valid values when PeriodType is set to Year: 1, 2, and 3.
MasterUserPassword	String	No	No	The password of the database account.	The password must be 6 to 32 characters in length and can contain letters, digits, and underscores (_).
AllocatePublicConnection	Boolean	No	No	Specifies whether to apply for a public endpoint for the instance.	None
PayType	String	No	No	The billing method of the instance.	Valid values: • Postpaid: pay-as-you-go • Prepaid: subscription
AutoPay	Boolean	No	No	Specifies whether to enable automatic payment for the instance.	Default value: False. Valid values: • True • False

DBMappings syntax

```
"DBMappings": [
  {
    "DBDescription": String,
    "CharacterSetName": String,
    "DBName": String
  }
]
```

DBMappings properties

Property	Type	Required	Editable	Description	Constraint
DBDescription	String	No	No	The description of the database.	The description must be 2 to 256 characters in length and can contain letters, digits, underscores (_), and hyphens (-). It must start with a letter and cannot start with `http://` or `https://`.

Property	Type	Required	Editable	Description	Constraint
CharacterSetName	String	Yes	No	The character set.	• Valid values when Engine is set to MySQL: utf8, gbk, latin1, and utf8mb4 (applicable to versions 5.5 and 5.6). • Valid values when Engine is set to SQLServer: Chinese_PRC_CI_AS, Chinese_PRC_CS_AS, SQL_Latin1_General_CP1_CI_AS, SQL_Latin1_General_CP1_CS_AS, and Chinese_PRC_BIN.
DBName	String	Yes	No	The name of the database.	The name must be unique. It can be up to 64 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter.

Response parameters

Fn::GetAtt

- InnerPort: the internal port of the instance.
- OrderId: the order ID of the instance.
- PublicConnectionString: the public endpoint of the instance.
- InnerIPAddress: the internal IP address of the instance.
- DBInstanceId: the ID of the instance.
- PublicIPAddress: the public IP address of the instance.
- PublicPort: the public port of the instance.
- InnerConnectionString: the internal endpoint of the instance.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "PeriodType": {
      "Type": "String",
      "Description": "Charge period for created instances.",
      "AllowedValues": [
        "Month",
        "Year"
      ],
      "Default": "Month"
    },
    "PrivateIpAddress": {
      "Type": "String",
      "Description": "The private ip for created instance."
    },
    "DBInstanceNetType": {
      "Type": "String",
```

```
      "Description": "Database instance net type, default is Intranet.Internet for public access, Intranet for private access.",
      "AllowedValues": [
        "Internet",
        "Intranet"
      ],
      "Default": "Intranet"
    },
    "AutoRenew": {
      "Type": "Boolean",
      "Description": "Auto renew the prepay instance. If the period type is by year, it will renew by year, else it will renew by month.",
      "AllowedValues": [
        "True",
        "true",
        "False",
        "false"
      ],
      "Default": false
    },
    "PreferredBackupPeriod": {
      "Type": "CommaDelimitedList",
      "Description": "Automate backups cycle if automated backups are enabled.",
      "AllowedValues": [
        "Monday",
        "Tuesday",
        "Wednesday",
        "Thursday",
        "Friday",
        "Saturday",
        "Sunday"
      ]
    },
    "DBInstanceStorage": {
      "Type": "Number",
      "Description": "Database instance storage size. mysql is [5,1000]. sql server 2008r2 is [10,1000], sql server 2012/2012_web/2016-web is [20,1000]. PostgreSQL and PPAS is [5,2000]. Increased every 5 GB, Unit in GB"
    },
    "CommodityCode": {
      "Type": "String",
      "Description": "The CommodityCode of the order.",
      "AllowedValues": [
        "rds",
        "bards",
        "rords"
      ],
      "Default": "rds"
    },
    "DBMappings": {
      "Type": "CommaDelimitedList",
      "Description": "Database mappings to attach to db instance."
    },
    "MultiAZ": {
      "Type": "Boolean",
      "Description": "Specifies if the database instance is a multiple Availability Zone deployment.",
      "AllowedValues": [
        "True",
```

```
        "true",
        "False",
        "false"
      ],
      "Default": false
    },
    "Engine": {
      "Type": "String",
      "Description": "Database instance engine type. Support MySQL/SQLServer/PostgreSQL/PPAS now.",
      "AllowedValues": [
        "MySQL",
        "SQLServer",
        "PostgreSQL",
        "PPAS"
      ]
    },
    "DBInstanceDescription": {
      "Type": "String",
      "Description": "Description of created database instance."
    },
    "Tags": {
      "Type": "Json",
      "Description": "The tags of an instance.\nYou should input the information of the tag with the format of the Key-Value, such as {\"key1\":\"value1\",\"key2\":\"value2\", ... \"key5\":\"value5\"}.\nAt most 5 tags can be specified.\nKey\nIt can be up to 64 characters in length.\nCannot begin with aliyun.\nCannot begin with http:// or https://.\nCannot be a null string.\nValue\nIt can be up to 128 characters in length.\nCannot begin with aliyun.\nCannot begin with http:// or https://.\nCan be a null string."
    },
    "EngineVersion": {
      "Type": "String",
      "Description": "Database instance version of the relative engine type.Support MySQL: 5.5/5.6/5.7; SQLServer: 2008r2, 20012, 2012_web, 2012_std_ha, 2012_ent_ha, 2016_web, 2016_std_ha, 2016_ent_ha; PostgreSQL:9.4; PPAS: 9.3.",
      "AllowedValues": [
        "5.5",
        "5.6",
        "5.7",
        "2008r2",
        "2012",
        "2012_web",
        "2012_std_ha",
        "2012_ent_ha",
        "2016_web",
        "2016_std_ha",
        "2016_ent_ha",
        "9.4",
        "9.3"
      ]
    },
    "ZoneId": {
      "Type": "String",
      "Description": "selected zone to create database instance. You cannot set the ZoneId parameter if the MultiAZ parameter is set to true."
    },
    "DBInstanceClass": {
      "Type": "String",
      "Description": "Database instance type. Refer the RDS database instance type reference, such as 'rds.mys2.large', 'rds.mss1.large', 'rds.pg.s1.small' etc"
    }
```

```
    },
    "AllocatePublicConnection": {
      "Type": "Boolean",
      "Description": "If true, allocate public connection automate.",
      "AllowedValues": [
        "True",
        "true",
        "False",
        "false"
      ]
    },
    "PreferredBackupTime": {
      "Type": "String",
      "Description": "The daily time range during which automated backups are created if automated ba
ckups are enabled.",
      "AllowedValues": [
        "00:00Z-01:00Z",
        "01:00Z-02:00Z",
        "02:00Z-03:00Z",
        "03:00Z-04:00Z",
        "04:00Z-05:00Z",
        "05:00Z-06:00Z",
        "06:00Z-07:00Z",
        "07:00Z-08:00Z",
        "08:00Z-09:00Z",
        "09:00Z-10:00Z",
        "10:00Z-11:00Z",
        "11:00Z-12:00Z",
        "12:00Z-13:00Z",
        "13:00Z-14:00Z",
        "14:00Z-15:00Z",
        "15:00Z-16:00Z",
        "16:00Z-17:00Z",
        "17:00Z-18:00Z",
        "18:00Z-19:00Z",
        "19:00Z-20:00Z",
        "20:00Z-21:00Z",
        "21:00Z-22:00Z",
        "22:00Z-23:00Z",
        "23:00Z-24:00Z"
      ]
    },
    "VSwitchId": {
      "Type": "String",
      "Description": "The vSwitch id of created instance. For VPC network, the property is required."
    },
    "Quantity": {
      "Type": "Number",
      "Description": "The number of instance to be created, default is 1, max number is 99",
      "MinValue": 1,
      "MaxValue": 99,
      "Default": 1
    },
    "Period": {
      "Type": "Number",
      "Description": "Prepaid time period. While choose by pay by month, it could be from 1 to 9. Whi
le choose pay by year, it could be from 1 to 3.",
      "MinValue": 1,
      "MaxValue": 9,
      "Default": 1
```

```
    },
    "MasterUserPassword": {
      "Type": "String",
      "Description": "The master password for the database instance. ",
      "MinLength": 8,
      "MaxLength": 32
    },
    "CouponCode": {
      "Type": "String",
      "Description": "The coupon code of the order."
    },
    "MasterUserType": {
      "Type": "String",
      "Description": "Privilege type of account.\n Normal: Common privilege. \n Super: High privilege
. And the default value is Normal.This parameter is valid for MySQL 5.5/5.6 only. MySQL 5.7, SQL Serv
er 2012/2016, PostgreSQL, and PPAS each can have only one initial account. \nOther accounts are creat
ed by the initial account that has logged on to the database.",
      "AllowedValues": [
        "Normal",
        "Super"
      ],
      "Default": "Normal"
    },
    "VpcId": {
      "Type": "String",
      "Description": "The VPC id of created database instance. For VPC network, the property is requi
red."
    },
    "MasterUsername": {
      "Type": "String",
      "Description": "The master user name for the database instance. "
    },
    "ConnectionMode": {
      "Type": "String",
      "Description": "Connection Mode for database instance,support 'Performance' and 'Safty' mode. D
efault is RDS system assigns. ",
      "AllowedValues": [
        "Performance",
        "Safty"
      ]
    },
    "BackupRetentionPeriod": {
      "Type": "Number",
      "Description": "The number of days for which automatic DB backups are retained.",
      "MinValue": 7,
      "MaxValue": 30,
      "Default": 7
    }
  },
  "Resources": {
    "PrepayDBInstance": {
      "Type": "ALIYUN::RDS::PrepayDBInstance",
      "Properties": {
        "PeriodType": {
          "Ref": "PeriodType"
        },
        "PrivateIpAddress": {
          "Ref": "PrivateIpAddress"
        },
```

```
        "DBInstanceNetType": {
          "Ref": "DBInstanceNetType"
        },
        "AutoRenew": {
          "Ref": "AutoRenew"
        },
        "PreferredBackupPeriod": {
          "Fn::Split": [
            ",",
            {
              "Ref": "PreferredBackupPeriod"
            },
            {
              "Ref": "PreferredBackupPeriod"
            }
          ]
        },
        "DBInstanceStorage": {
          "Ref": "DBInstanceStorage"
        },
        "CommodityCode": {
          "Ref": "CommodityCode"
        },
        "DBMappings": {
          "Fn::Split": [
            ",",
            {
              "Ref": "DBMappings"
            },
            {
              "Ref": "DBMappings"
            }
          ]
        },
        "MultiAZ": {
          "Ref": "MultiAZ"
        },
        "Engine": {
          "Ref": "Engine"
        },
        "DBInstanceDescription": {
          "Ref": "DBInstanceDescription"
        },
        "Tags": {
          "Ref": "Tags"
        },
        "EngineVersion": {
          "Ref": "EngineVersion"
        },
        "ZoneId": {
          "Ref": "ZoneId"
        },
        "DBInstanceClass": {
          "Ref": "DBInstanceClass"
        },
        "AllocatePublicConnection": {
          "Ref": "AllocatePublicConnection"
        },
        "PreferredBackupTime": {
          "Ref": "PreferredBackupTime"
```

```
        },
        "VSwitchId": {
          "Ref": "VSwitchId"
        },
        "Quantity": {
          "Ref": "Quantity"
        },
        "Period": {
          "Ref": "Period"
        },
        "MasterUserPassword": {
          "Ref": "MasterUserPassword"
        },
        "CouponCode": {
          "Ref": "CouponCode"
        },
        "MasterUserType": {
          "Ref": "MasterUserType"
        },
        "VpcId": {
          "Ref": "VpcId"
        },
        "MasterUsername": {
          "Ref": "MasterUsername"
        },
        "ConnectionMode": {
          "Ref": "ConnectionMode"
        },
        "BackupRetentionPeriod": {
          "Ref": "BackupRetentionPeriod"
        }
      }
    }
  },
  "Outputs": {
    "InnerConnectionString": {
      "Description": "DB instance connection url by Intranet.",
      "Value": {
        "Fn::GetAtt": [
          "PrepayDBInstance",
          "InnerConnectionString"
        ]
      }
    },
    "DBInstanceId": {
      "Description": "The instance id of created database instance.",
      "Value": {
        "Fn::GetAtt": [
          "PrepayDBInstance",
          "DBInstanceId"
        ]
      }
    },
    "InnerIPAddress": {
      "Description": "IP Address for created DB instance of Intranet.",
      "Value": {
        "Fn::GetAtt": [
          "PrepayDBInstance",
          "InnerIPAddress"
```

```
          ]
        }
      },
      "PublicConnectionString": {
        "Description": "DB instance connection url by Internet.",
        "Value": {
          "Fn::GetAtt": [
            "PrepayDBInstance",
            "PublicConnectionString"
          ]
        }
      },
      "PublicIPAddress": {
        "Description": "IP Address for created DB instance of Internet.",
        "Value": {
          "Fn::GetAtt": [
            "PrepayDBInstance",
            "PublicIPAddress"
          ]
        }
      },
      "OrderId": {
        "Description": "The order id list of created instance.",
        "Value": {
          "Fn::GetAtt": [
            "PrepayDBInstance",
            "OrderId"
          ]
        }
      },
      "PublicPort": {
        "Description": "Internet port of created DB instance.",
        "Value": {
          "Fn::GetAtt": [
            "PrepayDBInstance",
            "PublicPort"
          ]
        }
      },
      "InnerPort": {
        "Description": "Intranet port of created DB instance.",
        "Value": {
          "Fn::GetAtt": [
            "PrepayDBInstance",
            "InnerPort"
          ]
        }
      }
    }
  }
```

5.1.5.5. ROS

5.1.5.5.1. ALIYUN::ROS::WaitCondition

ALIYUN::ROS::WaitCondition is used to create an instance to process UserData messages.

Statement

```
{
  "Type": "ALIYUN::ROS::WaitCondition",
  "Properties": {
    "Count": Number,
    "Handle": String,
    "Timeout": Number
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
Handle	String	No	No	Reference ALIYUN::ROS::WaitConditionHandle.	None
Timeout	Number	Yes	No	The length of time to wait for UserData messages.	Valid values: 1 to 43200. Unit: seconds.
Count	Number	No.	True	The total number of messages to be received.	None

Response parameters

Fn::GetAtt

- Data: A JSON-serialized dictionary that contains the signal Data after the most recent stack creation or update.
- LastData: a JSON-serialized dictionary that contains the signal data before the most recent stack update.
- JoinedErrorData: a string consisting of the ErrorData signal data.
- JoinedLastErrorData: a string consisting of the LastErrorData signal data.
- ErrorData: a JSON-serialized dictionary that contains the error signal data after the most recent stack creation or update.
- Lasterprotodata: a JSON-serialized dictionary that contains the error signal data before the most recent stack update.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WaitCondition": {
      "Type": "ALIYUN::ROS::WaitCondition",
      "Properties": {
        "Handle": {
          "Ref": "WaitConHandle"
        },
        "Timeout": 5,
        "Count": 2
      }
    },
    "WaitConHandle": {
      "Type": "ALIYUN::ROS::WaitConditionHandle"
    }
  },
  "Outputs": {
    "CurlCli": {
      "Value": {
        "Fn::GetAtt": [
          "WaitConHandle",
          "CurlCli"
        ]
      }
    },
    "Data": {
      "Value": {
        "Fn::GetAtt": [
          "WaitCondition",
          "Data"
        ]
      }
    }
  }
}
```

5.1.5.5.2. ALIYUN::ROS::WaitConditionHandle

ALIYUN::ROS::WaitConditionHandle is used to create an instance that sends and receives messages during UserData execution.

Statement

```
{
  "Type": "ALIYUN::ROS::WaitConditionHandle",
  "Properties": {
    "Count": Integer,
    "Mode": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
Count	Integer	No.	True	The total number of messages to be received.	Default value: -1.
Mode	String	Yes	True	If you set this parameter to Increment, all previous signals will be updated before they are deleted. If you set this parameter to Full, no previous signals will be deleted unless the Count parameter is specified.	Valid values: • Increment • Full Default value: Full.

Response parameters

Fn::GetAtt

- CurlCli: A curl Command is generated by the resource. You can use the command to send the UserData execution result or status to Resource Orchestration Service.
- WindowsCurlCli: provides Windows with cURL CLI command prefixes and sends a message indicating that the execution is completed or failed. Windows does not support the curl command. Therefore, you must install curl.exe and add it to PATH. You can add `--data-binary "{\"status\": \" success \"}` to indicate success, or by adding `--data-binary "{\"status\": \" failure \"}` to indicate failure.
- PowerShellCurlCli: provides PowerShell with cURL CLI command prefixes and sends a message indicating that the execution is completed or failed. Because this cmdlet was introduced in PowerShell 3.0, make sure that the PowerShell version meets this constraint. By `$PSVersionTable.PSVersion` displays the version. You can add `-Body '{"status": "success "}` to indicate success, or by adding `-Body '{"status": "failure "}` to indicate failure.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "Mode": {
      "Type": "String",
      "Description": "If set to Increment, all old signals will be deleted before update. In this mode, WaitCondition.Count should reference an incremental value instead of a full value, such as ScalingGroupEnable.ScalingRuleArisExecuteResultNumberOfAddedInstances.\n\nIf set to Full, no old signal will be deleted unless Count is set. In this mode, WaitCondition.Count should reference a full value, such as the same value with InstanceGroup.MaxAmount. It is recommended to use this mode with Count.\n\nDefault to Full.",
      "AllowedValues": [
        "Increment",
        "Full"
      ],
      "Default": "Full"
    },
    "Count": {
```

```
      "Type": "Number",
      "Description": "There are 3 preconditions that make Count taking effect:\n1.Mode is set to Full
.\n2.Count >= 0.\n3.The id of signal is not specified. If so, it will be a self-increasing integer st
arted from 1. For example, the id of the first signal is 1, the id of the second signal is 2, and so
on.\n\nIf Count takes effect, signals with id > Count will be deleted before update.\nThe default val
ue is -1, which means no effect.\nIt is recommended to quote the same value with WaitCondition.Count.
",
      "Default": -1
    }
  },
  "Resources": {
    "WaitConditionHandle": {
      "Type": "ALIYUN::ROS::WaitConditionHandle",
      "Properties": {
        "Mode": {
          "Ref": "Mode"
        },
        "Count": {
          "Ref": "Count"
        }
      }
    }
  },
  "Outputs": {
    "CurlCli": {
      "Description": "Convenience attribute, provides curl CLI command prefix, which can be used for
signalling handle completion or failure.  You can signal success by adding --data-binary '{\"status\"
: \"SUCCESS\"}' , or signal failure by adding --data-binary '{\"status\": \"FAILURE\"}'",
      "Value": {
        "Fn::GetAtt": [
          "WaitConditionHandle",
          "CurlCli"
        ]
      }
    },
    "WindowsCurlCli": {
      "Description": "Convenience attribute, provides curl CLI command prefix for Windows, which can
be used for signalling handle completion or failure. As Windows does not support curl command, you ne
ed to install curl.exe and add it to PATH first. You can signal success by adding --data-binary \"{\\
\"status\\\": \\\"SUCCESS\\\"}\" , or signal failure by adding --data-binary \"{\\\"status\\\": \\\"F
AILURE\\\"}\" ",
      "Value": {
        "Fn::GetAtt": [
          "WaitConditionHandle",
          "WindowsCurlCli"
        ]
      }
    },
    "PowerShellCurlCli": {
      "Description": "Convenience attribute, provides curl CLI command prefix for PowerShell, which c
an be used for signalling handle completion or failure. As this cmdlet was introduced in PowerShell 3
.0, ensure the version of PowerShell satisfies the constraint. (Show the version via $PSVersionTable.
PSVersion.) You can signal success by adding -Body '{\"status\": \"SUCCESS\"}' , or signal failure by
adding -Body '{\"status\": \"FAILURE\"}' ",
      "Value": {
        "Fn::GetAtt": [
          "WaitConditionHandle",
          "PowerShellCurlCli"
        ]
```

```
        }
      }
    }
}
```

5.1.5.5.3. ALIYUN::ROS::Stack

ALIYUN::ROS::Stack is used to create a nested stack. You can have a maximum of five nested levels.

ALIYUN::ROS::Stack is used in a top-level template to nest stacks as resources.

You can add output values from a nested stack contained within the template. You can use Fn::GetAtt together with the logical name of the nested stack and the output name in the Outputs.NestedStackOutputName format.

> **Note** We recommend that you run an update to the Nested stack from the parent stack.

When you apply a template change to update a top-level stack, ROS updates the top-level stack and initiates an update to its nested stacks. Resource orchestration service (ROS) updates resources that have been modified in the nested stack, but does not update resources that have not been modified in the nested stack.

Statement

```
{
  "Type": "ALIYUN::ROS::Stack",
  "Properties": {
    "TemplateURL": String,
    "TimeoutMins": Number,
    "Parameters": Map
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint

Parameter	Type	Required	Editable	Description	Constraint
TemplateURL	String	No	Yes	The URL of the file containing the template body. The template file can be up to 524,288 bytes in size. The URL must point to a template located on the http or https Web server or Alibaba Cloud OSS bucket. For example: `oss://ros/template/demo`, `oss://ros/template/demo?RegionId=cn-hangzhou`. If the region of the OSS bucket is not specified, the RegionId of the stack is used.	The URL can be up to 1,024 bytes in length.
TimeoutMins	Number	No.	True	The length of time that ROS will wait for the nested stack to be created or updated.	Unit: minutes. Default value: 60.

Parameter	Type	Required	Editable	Description	Constraint
Parameters	Map	No.	True	A set of value pairs that represent the parameters passed to ROS when this Nested stack is created. Each parameter has a name corresponding to a parameter defined in the embedded template and the value to which you want to set the parameter. This parameter is required if the nested stack needs input parameters.	None

Response parameters

Fn::GetAtt

You can use the following code to obtain the output of the nested stack:

```
{
  "Fn::GetAtt": [
    "<nested_stack>",
    "Outputs.<nested_stack_output_name>"
  ]
}
```

When you use `Ref` to reference resources in a nested stack, the Alibaba Cloud Resource Name (ARN) of the nested stack is returned. Example: `arn:acs:ros::cn-hangzhou:12345****:stacks/test-nested-stack-Demo-jzkyq7mn2ykj/e71c1e04-1a57-46fc-b9a4-cf7ce0d3****`

Examples

- The following code provides an example of how to create a VPC, a VSwitch, and a security group in a nested stack and save the output results to the *oss://ros/template/vpc.txt* directory:

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Description": "One VPC, VSwitch, security group.",
  "Parameters": {
    "ZoneId": {
      "Type": "String",
      "Description": "The available zone"
    },
    "SecurityGroupName": {
      "Type": "String",
      "Description": "The security group name",
```

```
      "Default": "my-sg-name"
    },
    "VpcName": {
      "Type": "String",
      "Description": "The VPC name",
      "MinLength": 2,
      "MaxLength": 128,
      "ConstraintDescription": "[2, 128] English or Chinese letters",
      "Default": "my-vpc-name"
    },
    "VpcCidrBlock": {
      "Type": "String",
      "AllowedValues": [
        "192.168.0.0/16",
        "172.16.0.0/12",
        "10.0.0.0/8"
      ],
      "Default": "10.0.0.0/8"
    },
    "VSwitchCidrBlock": {
      "Type": "String",
      "Description": "The VSwitch subnet which must be within VPC",
      "Default": "10.0.10.0/24"
    },
    "UpdateVersion": {
      "Type": "Number",
      "Default": 0
    }
  },
  "Resources": {
    "Vpc": {
      "Type": "ALIYUN::ECS::VPC",
      "Properties": {
        "CidrBlock": {
          "Ref": "VpcCidrBlock"
        },
        "VpcName": {
          "Ref": "VpcName"
        }
      }
    },
    "VSwitch": {
      "Type": "ALIYUN::ECS::VSwitch",
      "Properties": {
        "CidrBlock": {
          "Ref": "VSwitchCidrBlock"
        },
        "ZoneId": {
          "Ref": "ZoneId"
        },
        "VpcId": {
          "Fn::GetAtt": [
            "Vpc",
            "VpcId"
          ]
        }
      }
    },
    "SecurityGroup": {
      "Type": "ALIYUN::ECS::SecurityGroup",
```

```
      "Properties": {
        "SecurityGroupName": {
          "Ref": "SecurityGroupName"
        },
        "VpcId": {
          "Ref": "Vpc"
        }
      }
    },
    "WaitConditionHandle": {
      "Type": "ALIYUN::ROS::WaitConditionHandle",
      "Properties": {
        "UpdateVersion": {
          "Ref": "UpdateVersion"
        }
      }
    }
  },
  "Outputs": {
    "SecurityGroupId": {
      "Value": {
        "Fn::GetAtt": [
          "SecurityGroup",
          "SecurityGroupId"
        ]
      }
    },
    "VpcId": {
      "Value": {
        "Fn::GetAtt": [
          "Vpc",
          "VpcId"
        ]
      }
    },
    "VSwitchId": {
      "Value": {
        "Fn::GetAtt": [
          "VSwitch",
          "VSwitchId"
        ]
      }
    }
  }
}
```

- The following code provides an example of a top-level stack:

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Description": "One ECS instance.",
  "Parameters": {
    "ImageId": {
      "Default": "centos_7",
      "Type": "String",
      "Description": "Image Id, represents the image resource to startup the ECS instance"
    },
    "InstanceType": {
      "Type": "String",
      "Description": "The ECS instance type."
```

```
      "Description": "The ECS instance type",
      "Default": "ecs.xn4.small"
    },
    "ZoneId": {
      "Type": "String",
      "Description": "The available zone "
    },
    "InstanceChargeType": {
      "Type": "String",
      "AllowedValues": [
        "PrePaid",
        "PostPaid"
      ],
      "Default": "PostPaid",
      "Description": "The instance charge type"
    },
    "SecurityGroupName": {
      "Type": "String",
      "Description": "The security group name",
      "Default": "my-sg-name"
    },
    "NetworkInterfaceName": {
      "Type": "String",
      "Description": "The Network interface name",
      "Default": "my-eni-name"
    },
    "VpcName": {
      "Type": "String",
      "Description": "The VPC name",
      "MinLength": 2,
      "MaxLength": 128,
      "ConstraintDescription": "[2, 128] English or Chinese letters",
      "Default": "my-vpc-name"
    },
    "IoOptimized": {
      "AllowedValues": [
        "none",
        "optimized"
      ],
      "Description": "IO optimized, optimized is for the IO optimized instance type",
      "Type": "String",
      "Default": "optimized"
    },
    "SystemDiskCategory": {
      "AllowedValues": [
        "cloud",
        "cloud_efficiency",
        "cloud_ssd"
      ],
      "Description": "System disk category: average cloud disk(cloud), efficient cloud disk(cloud_
efficiency) or SSD cloud disk(cloud_ssd)",
      "Type": "String",
      "Default": "cloud_ssd"
    },
    "VpcCidrBlock": {
      "Type": "String",
      "AllowedValues": [
        "192.168.0.0/16",
        "172.16.0.0/12",
        "10.0.0.0/8"
```

```
      ],
      "Default": "10.0.0.0/8"
    },
    "VSwitchCidrBlock": {
      "Type": "String",
      "Description": "The VSwitch subnet which must be within VPC",
      "Default": "10.0.10.0/24"
    },
    "UpdateVersion": {
      "Type": "Number",
      "Default": 0
    }
  },
  "Resources": {
    "NetworkStack": {
      "Type": "ALIYUN::ROS::Stack",
      "Properties": {
        "TemplateURL": "oss://ros/template/vpc.txt",
        "TimeoutMins": 5,
        "Parameters": {
          "ZoneId": {
            "Ref": "ZoneId"
          },
          "SecurityGroupName": {
            "Ref": "SecurityGroupName"
          },
          "VpcName": {
            "Ref": "VpcName"
          },
          "VpcCidrBlock": {
            "Ref": "VpcCidrBlock"
          },
          "VSwitchCidrBlock": {
            "Ref": "VSwitchCidrBlock"
          },
          "UpdateVersion": {
            "Ref": "UpdateVersion"
          }
        }
      }
    },
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId": {
          "Ref": "ImageId"
        },
        "InstanceType": {
          "Ref": "InstanceType"
        },
        "InstanceChargeType": {
          "Ref": "InstanceChargeType"
        },
        "SecurityGroupId": {
          "Fn::GetAtt": [
            "NetworkStack",
            "Outputs.SecurityGroupId"
          ]
        },
```

```
        "VpcId": {
          "Fn::GetAtt": [
            "NetworkStack",
            "Outputs.VpcId"
          ]
        },
        "VSwitchId": {
          "Fn::GetAtt": [
            "NetworkStack",
            "Outputs.VSwitchId"
          ]
        },
        "IoOptimized": {
          "Ref": "IoOptimized"
        },
        "ZoneId": {
          "Ref": "ZoneId"
        },
        "SystemDisk_Category": {
          "Ref": "SystemDiskCategory"
        },
        "DiskMappings": [
          {
            "Category": "cloud_ssd",
            "Size": 20
          }
        ]
      }
    }
  },
  "Outputs": {
    "InstanceId": {
      "Value": {
        "Fn::GetAtt": [
          "WebServer",
          "InstanceId"
        ]
      }
    },
    "PublicIp": {
      "Value": {
        "Fn::GetAtt": [
          "WebServer",
          "PublicIp"
        ]
      }
    },
    "SecurityGroupId": {
      "Value": {
        "Fn::GetAtt": [
          "NetworkStack",
          "Outputs.SecurityGroupId"
        ]
      }
    },
    "VpcId": {
      "Value": {
        "Fn::GetAtt": [
          "NetworkStack",
          "Outputs.VpcId"
```

```
        ]
      }
    },
    "VSwitchId": {
      "Value": {
        "Fn::GetAtt": [
          "NetworkStack",
          "Outputs.VSwitchId"
        ]
      }
    },
    "NetworkStackArn": {
      "Value": {
        "Ref": "NetworkStack"
      }
    }
  }
}
```

5.1.5.6. SLB

5.1.5.6.1. ALIYUN::SLB::AccessControl

ALIYUN::SLB::AccessControl is used to create an access control list (ACL).

Syntax

```
{
  "Type": "ALIYUN::SLB::AccessControl",
  "Properties": {
    "AddressIPVersion": String,
    "AclName": String,
    "AclEntrys": List
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
AddressIPVersion	String	No	No	The Internet protocol version.	Valid values: ipv4 and ipv6.
AclName	String	Yes	Yes	The name of the ACL.	None
AclEntrys	List	No	No	The list of ACL entries.	A list can contain up to 50 ACL entries.

AclEntrys syntax

```
"AclEntrys": [
  {
    "comment": String,
    "entry": String
  }
]
```

AclEntrys properties

Property	Type	Required	Editable	Description	Constraint
comment	String	No	No	The comments on ACL entries.	None
entry	String	Yes	No	The authorized IP addresses or CIDR blocks.	None

Response parameters

Fn::GetAtt

AclId: the ID of the ACL.

Examples

Resource usage example

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "AccessControl": {
      "Type": "ALIYUN::SLB::AccessControl",
      "Properties": {
        "AddressIPVersion": {
          "Ref": "AddressIPVersion"
        },
        "AclName": {
          "Ref": "AclName"
        },
        "AclEntrys": {
          "Fn::Split": [",", {
            "Ref": "AclEntrys"
          }, {
            "Ref": "AclEntrys"
          }]
        }
      }
    }
  },
  "Parameters": {
    "AddressIPVersion": {
      "Type": "String",
      "Description": "IP version. Could be \"ipv4\" or \"ipv6\".",
      "AllowedValues": ["ipv4", "ipv6"]
    },
    "AclName": {
      "Type": "String",
      "Description": "The name of the access control list."
    },
    "AclEntrys": {
      "Type": "CommaDelimitedList",
      "Description": "A list of acl entrys. Each entry can be IP addresses or CIDR blocks. Max length
: 50.",
      "MaxLength": 50
    }
  },
  "Outputs": {
    "AclId": {
      "Description": "The ID of the access control list.",
      "Value": {
        "Fn::GetAtt": ["AccessControl", "AclId"]
      }
    }
  }
}
```

Example of combined use of SLB-related resources

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "LoadBalancer": {
      "Type": "ALIYUN::SLB::LoadBalancer",
      "Properties": {
        "LoadBalancerName": "slb-with-listener-and-acl",
        "AddressType": "internet"
```

```
        "InternetChargeType": "paybybandwidth",
        "Bandwidth": 10,
        "VpcId": "vpc-xxxxxxxxxxxxxxxxxxx",
        "VSwitchId": "vsw-xxxxxxxxxxxxxxxxxxx"
      }
    },
    "ACL": {
      "Type": "ALIYUN::SLB::AccessControl",
      "Properties": {
        "AclName": "acl-for-listener",
        "AddressIPVersion": "ipv4",
        "AclEntrys": [
          {
            "entry": "192.168.x.x"
          },
          {
            "entry": "10.0.x.x/24",
            "comment": "just comment"
          }
        ]
      }
    },
    "CreateListener": {
      "Type": "ALIYUN::SLB::Listener",
      "Properties": {
        "LoadBalancerId": {
          "Ref": "LoadBalancer"
        },
        "ListenerPort": "80",
        "BackendServerPort": 8080,
        "Bandwidth": 1,
        "Protocol": "http",
        "HealthCheck": {
          "HealthyThreshold": 3,
          "UnhealthyThreshold": 3,
          "Interval": 2,
          "Timeout": 5
        },
        "Scheduler": "wrr",
        "RequestTimeout": 179,
        "IdleTimeout": 59,
        "AclId": {
          "Ref": "ACL"
        },
        "AclStatus": "on",
        "AclType": "white"
      }
    }
  },
  "Outputs": {
    "LoadBalanceDetails": {
      "Value": {
        "Fn::GetAtt": [
          "LoadBalancerId",
          "Listeners"
        ]
      }
    }
  }
```

```
}
```

5.1.5.6.2. ALIYUN::SLB::BackendServerAttachment

ALIYUN::SLB::BackendServerAttachment is used to add backend servers.

Statement

```
{
  "Type": "ALIYUN::SLB::BackendServerAttachment",
  "Properties": {
    "LoadBalancerId": String,
    "BackendServers": List,
    "BackendServerList": List,
    "BackendServerWeightList": List
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
LoadBalancerId	String	No	No	The unique ID of the SLB instance.	None
BackendServerList	List	No.	True	The list of backend servers to add.	You can call this operation with LoadBalancerId and BackendServerWeightList. Separate ECS instance IDs with commas (,). This parameter is ignored when the BackendServers parameter is specified.

Parameter	Type	Required	Editable	Description	Constraint
BackendServerWeightList	List	No.	True	The weights of the ECS instances in the BackendServerList, which are specified in order.	If this parameter is not specified, the weight of all ECS instances included in the BackendServerList is 100. When the BackendServerWeightList length is less than BackendServerList, the last value in the BackendServerWeightList is used to weight the remaining ECS instances in the BackendServerList.
BackendServers	List	No.	True	The list of backend servers to add.	Only backend servers in the running state can be attached to the SLB instance.

BackendServers syntax

```
"BackendServers": [
  {
    "ServerId" : String,
    "Weight" : Integer
  }
]
```

BackendServers properties

Parameter	Type	Required	Editable	Description	Constraint
ServerId	String	No	Yes	The ID of the ECS instance that acts as a backend server.	The ECS instance must be in the Running state.
Weight	Integer	Retained	Yes	The weight of the ECS instance in the SLB instance.	Valid values: 0 to 100. Default value: 100.

Response parameters

Fn::GetAtt

- BackendServers: the backend servers added to the SLB instance.

- LoadBalancerId: the ID of the SLB instance.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Attachment2": {
      "Type": "ALIYUN::SLB::BackendServerAttachment",
      "Properties": {
        "LoadBalancerId": "15187200816-cn-beijing-btc-****",
        "BackendServerList": [
          "i-25o0m****",
          "i-25zsk****"
        ],
        "BackendServerWeightList": [
          "20",
          "100"
        ]
      }
    }
  }
}
```

5.1.5.6.3. ALIYUN::SLB::BackendServerToVServerGroupAddition

ALIYUN::SLB::BackendServerToVServerGroupAddition is used to add backend servers to an existing VServer group.

Statement

```
{
  "Type": "ALIYUN::SLB::BackendServerToVServerGroupAddition",
  "Properties": {
    "BackendServers": List,
    "VServerGroupId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
VServerGroupId	String	No	No	The ID of the VServer group.	None
BackendServers	List	Retained	Yes	The list of ECS instances to be added.	None

BackendServers syntax

```
"BackendServers": [
  {
    "ServerId": String,
    "Port": Integer,
    "Weight": Integer
  }
]
```

BackendServers properties

Parameter	Type	Required	Editable	Description	Constraint
ServerId	String	No	Yes	The ID of the ECS instance that acts as a backend server.	None
Port	Integer	Retained	Yes	The ECS port number that is listened to in the server load balancer instance.	Valid values: 1 to 65535.
Weight	Integer	Retained	Yes	The weight of the ECS instance to be attached to the SLB instance.	Valid values: 0 to 100.

Response parameters

Fn::GetAtt

VServerGroupId: the ID of the VServer group.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "AttachVServerGroup": {
      "Type": "ALIYUN::SLB::BackendServerToVServerGroupAddition",
      "Properties": {
        "VServerGroupId": "sg-2zenh4ndwrqg14yt0****",
        "BackendServers": [
          {
            "ServerId": "i-25zsk****",
            "Weight": 20,
            "Port": 8080
          },
          {
            "ServerId": "i-25zsk****",
            "Weight": 100,
            "Port": 8081
          }
        ]
      }
    }
  }
}
```

5.1.5.6.4. ALIYUN::SLB::Certificate

ALIYUN::SLB::Certificate is used to upload a certificate to an SLB instance. Server certificates and CA certificates are supported.

Notice

- You can upload only one CA certificate at a time ("CertificateType": "CA ").
- You can upload only one server certificate and the corresponding private key at a time ("CertificateType": "Server ").

Syntax

```
{
  "Type": "ALIYUN::SLB::Certificate",
  "Properties": {
    "CertificateName": String,
    "Certificate": String,
    "AliCloudCertificateName": String,
    "PrivateKey": String,
    "ResourceGroupId": String,
    "CertificateType": String,
    "AliCloudCertificateId": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint

Property	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	No	No	The ID of the resource group.	None
CertificateName	String	No	Yes	The name of the certificate.	None
Certificate	String	Yes	No	The public key of the certificate.	None
AliCloudCertificateName	String	No	No	The name of the Alibaba Cloud certificate.	None
PrivateKey	String	No	No	The server private key that you want to upload.	None
AliCloudCertificateId	String	No	No	The ID of the Alibaba Cloud certificate.	This parameter is required if you use a certificate from Alibaba Cloud SSL Certificates Service.
CertificateType	String	No	No	The type of the certificate.	Valid values: Server and CA.

Response parameters

Fn::GetAtt

- CertificateId: the ID of the certificate.
- Fingerprint: the fingerprint of the certificate.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "CertificateType": {
      "Type": "String",
      "Description": "The type of the certificate.",
      "AllowedValues": [
        "Server",
        "CA"
      ],
      "Default": "Server"
    },
    "AliCloudCertificateName": {
      "Type": "String",
      "Description": "The name of the Alibaba Cloud certificate."
    },
    "PrivateKey": {
      "Type": "String",
      "Description": "The private key."
```

```
    },
    "CertificateName": {
      "Type": "String",
      "Description": "The name of the certificate."
    },
    "Certificate": {
      "Type": "String",
      "Description": "The content of the certificate public key."
    },
    "AliCloudCertificateId": {
      "Type": "String",
      "Description": "The ID of the Alibaba Cloud certificate."
    }
  },
  "Resources": {
    "Certificate": {
      "Type": "ALIYUN::SLB::Certificate",
      "Properties": {
        "CertificateType": {
          "Ref": "CertificateType"
        },
        "AliCloudCertificateName": {
          "Ref": "AliCloudCertificateName"
        },
        "PrivateKey": {
          "Ref": "PrivateKey"
        },
        "CertificateName": {
          "Ref": "CertificateName"
        },
        "Certificate": {
          "Ref": "Certificate"
        },
        "AliCloudCertificateId": {
          "Ref": "AliCloudCertificateId"
        }
      }
    }
  },
  "Outputs": {
    "Fingerprint": {
      "Description": "The fingerprint of the certificate.",
      "Value": {
        "Fn::GetAtt": [
          "Certificate",
          "Fingerprint"
        ]
      }
    },
    "CertificateId": {
      "Description": "The ID of the certificate.",
      "Value": {
        "Fn::GetAtt": [
          "Certificate",
          "CertificateId"
        ]
      }
    }
  }
}
```

```
}
```

5.1.5.6.5. ALIYUN::SLB::DomainExtension

ALIYUN::SLB::DomainExtension is used to create a domain extension for an SLB instance.

Statement

```
{
  "Type": "ALIYUN::SLB::DomainExtension",
  "Properties": {
    "Domain": String,
    "ListenerPort": Integer,
    "ServerCertificateId": String,
    "LoadBalancerId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
Domain	String	No	No	The custom domain name.	None
ListenerPort	Integer	Yes	No	The frontend port used by the HTTPS listener of the SLB instance.	Valid values: 1 to 65535.
ServerCertificateId	String	No	Yes	The ID of the certificate corresponding to the domain name.	None
LoadBalancerId	String	No	No	The ID of the SLB instance.	None

Response parameters

Fn::GetAtt

- DomainExtensionId: the ID of the created domain extension.
- ListenerPort: The frontend port used by the SLB instance.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "DomainExtension": {
      "Type": "ALIYUN::SLB::DomainExtension",
      "Properties": {
        "Domain": "*.example.com",
        "ListenerPort": "443",
        "ServerCertificateId": "123157908552****_166f8204689_1714763408_70998****",
        "LoadBalancerId": "lb-bp1o94dp5i6earr9g****"
      }
    }
  },
  "Outputs": {
    "DomainExtensionId": {
      "Value": {
        "Fn::GetAtt": [
          "DomainExtension",
          "DomainExtensionId"
        ]
      }
    },
    "ListenerPort": {
      "Value": {
        "Fn::GetAtt": [
          "DomainExtension",
          "ListenerPort"
        ]
      }
    }
  }
}
```

5.1.5.6.6. ALIYUN::SLB::Listener

ALIYUN::SLB::Listener is used to create a listener for an SLB instance.

Statement

```
{
  "Type": "ALIYUN::SLB::Listener",
  "Properties": {
    "MasterSlaveServerGroupId": String,
    "AclStatus": String,
    "Protocol": String,
    "AclId": String,
    "ServerCertificateId": String,
    "HealthCheck": Map,
    "RequestTimeout": Integer,
    "IdleTimeout": Integer,
    "ListenerPort": Integer,
    "HttpConfig": Map,
    "Bandwidth": Integer,
    "AclType": String,
    "BackendServerPort": Integer,
    "Scheduler": String,
    "LoadBalancerId": String,
    "CACertificateId": String,
    "Persistence": Map,
    "VServerGroupId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
MasterSlaveServerGroupId	String	Yes	Released	The ID of the active/standby server group.	None
AclStatus	String	Yes	Released	Specifies whether to enable access control on the listener.	Valid values: • on • off Default value: off.
AclId	String	Yes	Released	The ID of the access control list (ACL) to which the listener is bound. This parameter is required when the AclStatus parameter is set to on.	None
				The type of the ACL. Valid values: white and black. • white: specifies the ACL as a whitelist. Only	

Parameter	Type	Required	Editable	Description	Constraint
AclType	String	Yes	Released	requests from the IP addresses or CIDR blocks specified in the ACL are forwarded. Whitelists are applicable to scenarios where you want an application to only be accessed from specific IP addresses. Configuring a whitelist poses risks to your services. After a whitelist is configured, only the IP addresses specified in the whitelist are able to access the SLB listener. If a whitelist is enabled without any IP addresses specified, the SLB listener will not forward any requests. • black: specifies the ACL as a blacklist. Requests from the IP addresses or CIDR blocks specified in the ACL are not forwarded. Blacklists are applicable to scenarios where you want an application to only be denied access from specific IP addresses.	Valid values: • White • Black

Parameter	Type	Required	Editable	Description	Constraint
				If a blacklist is enabled without any IP addresses specified, the SLB listener will forward all requests. This parameter is required when the AclStatus parameter is set to on.	
Protocol	String	No	No	The Internet protocol over which the listener will forward requests.	Valid values: • http • https • tcp • udp
ListenerPort	Integer	Yes	No	The frontend port used by the SLB instance.	Valid values: 1 to 65535.

Parameter	Type	Required	Editable	Description	Constraint
Bandwidth	Integer	Yes	No	The peak bandwidth of the listener. Unit: Mbit/s.	• Valid values: -1 and 1 to 1000. • For an SLB instance that is connected to the Internet and billed by fixed bandwidth, this parameter cannot be set to -1, and the sum of peak bandwidth values assigned to different listeners cannot exceed the Bandwidth value specified when the SLB instance is created. For an SLB instance that is connected to the Internet and billed by traffic, this parameter can be set to -1. Unit: Mbit/s.
BackendServerPort	Integer	Yes	No	The backend port used by the SLB instance.	Valid values: 1 to 65535.
LoadBalancerId	String	No	No	The ID of the SLB instance.	None
HealthCheck	Map	Erased	Released	The health check settings of the listener.	None
Persistence	Map	Erased	Released	The persistence properties.	None

Parameter	Type	Required	Editable	Description	Constraint
Scheduler	String	Yes	Released	The algorithm used to direct traffic to individual servers.	Valid values: • wrr • wlc Default value: wrr
CACertificateId	String	Yes	Released	The ID of the CA certificate.	Only valid for. HTTPS
ServerCertificateId	String	Yes	Released	The ID of the server certificate.	This parameter is required and valid only for HTTPS listeners.
VServerGroupId	String	Yes	Released	The ID of the VServer group.	None
RequestTimeout	String	Optional	Released	The request timeout period. Unit: seconds.	Valid values: 1 to 180.
IdleTimeout	String	Optional	Released	The idle connection timeout period. Unit: seconds.	Valid values: 1 to 60.
HttpConfig	Map	Erased	Released	The HTTP configurations.	None

HealthCheck syntax

```
"HealthCheck": {
  "Domain": String,
  "Interval": Integer,
  "URI": String,
  "HttpCode": String,
  "HealthyThreshold": Integer,
  "Timeout": Integer,
  "UnhealthyThreshold": Integer,
  "Port": Integer
}
```

HealthCheck properties

Parameter	Type	Required	Editable	Description	Constraint

Parameter	Type	Required	Editable	Description	Constraint
Domain	String	Yes	Released	The domain name used for health checks.	• The value can be $_ip, a custom string, or an empty string. • A custom string must be 1 to 80 characters in length and can contain only letters, digits, hyphens (-), and periods (.). • When this parameter is set to $_ip or left empty, the SLB instance uses the private IP addresses of backend servers as the domain names for health checks.
Interval	String	Optional	Released	The time interval between consecutive health checks. Unit: seconds.	Valid values: 1 to 5. Unit: seconds.

Parameter	Type	Required	Editable	Description	Constraint
URI	String	Yes	Released	The URI used for health checks.	• The URI must be 1 to 80 characters in length • and can contain letters, digits, hyphens (-), forward slashes (/), periods (.), percent signs (%), question marks (?), number signs (#), and ampersands (&). It must start with a forward slash (/).
HttpCode	String	Yes	Released	The HTTP status code that indicates a positive health status of the backend servers.	• Valid values: http_2xx, http_3xx, http_4xx, and http_5xx. • Separate multiple HTTP status codes with commas (,). Default value: http_2xx
HealthyThreshold	String	Optional	Released	The threshold used to determine that the backend servers are healthy. This value indicates the number of consecutive successful health checks required before the health status of a backend server can be changed from fail to success.	Valid values: 1 to 10.

Parameter	Type	Required	Editable	Description	Constraint
Timeout	String	Optional	Released	The length of time to wait for the response from a health check. Unit: seconds.	Valid values: 1 to 50. **Notice** This parameter is valid only when its value is greater than or equal to that of the Interval parameter. Otherwise, this parameter will be overridden by the Interval value.
UnhealthyThreshold	String	Optional	Released	The threshold used to determine that the backend servers are unhealthy. This value indicates the number of consecutive failed health checks required before the health status of a backend server can be changed from success to fail.	Valid values: 1 to 10.
Port	String	Optional	Released	The port used for health checks.	Valid values: 0 to 65535.

Persistence syntax

```
"Persistence": {
  "PersistenceTimeout": Integer,
  "CookieTimeout": Integer,
  "XForwardedFor": String,
  "Cookie": String,
  "StickySession": String,
  "StickySessionType": String
}
```

Persistence properties

Parameter	Type	Required	Editable	Description	Constraint
StickySession	String	No	No	Specifies whether to enable session persistence.	Valid values: • on • off
PersistenceTimeout	String	Optional	Released	The maximum duration that the client can be connected to the server. Unit: seconds.	Valid values: 0 to 1000. The default value is 0, which indicates that connection persistence is disabled. Unit: seconds.
CookieTimeout	String	Optional	Released	The maximum duration the cookie can be retained before it expires. Unit: seconds.	This parameter is required when the StickySession parameter is set to on and the StickySessionType parameter is set to insert. Valid values: 1 to 86400. Unit: seconds.
XForwardedFor	String	Yes	Released	Specifies whether to use the X-Forwarded-For header field to obtain the real IP address of the client.	Valid values: • on • off Default value: on

Parameter	Type	Required	Editable	Description	Constraint
Cookie	String	Yes	Released	The cookie configured on the backend server.	• The parameter value must be 1 to 200 characters in length and follow the RFC 2965 standard. It can contain only ASCII characters. It cannot contain commas (,), semicolons (;), or spaces, and cannot start with a dollar sign ($). • This parameter is required when the StickySession parameter is set to on and the StickySession Type parameter is set to server.

Parameter	Type	Required	Editable	Description	Constraint
StickySessionType	String	Yes	Released	The method for processing cookies.	• Valid values: insert and server. • When this parameter is set to insert, SLB adds a cookie to the first response from the backend server. Then, the next request contains the cookie and the listener distributes the request to the same backend server. When this parameter is set to server, SLB overwrites the original cookie when a new cookie is set. The next time the client carries the new cookie to access SLB, the listener distributes the request to the previously recorded backend server. • This parameter is required when the StickySession parameter is set to on. This parameter is ignored when the StickySession parameter is set to off.

HttpConfig syntax

```
"HttpConfig": {
  "ForwardPort": Integer,
  "ListenerForward": String
}
```

HttpConfig properties

Parameter	Type	Required	Editable	Description	Constraint
ForwardPort	String	Optional	Released	The port used to redirect HTTP requests to HTTPS.	Valid values: 1 to 65535. Default value: 443.
ListenerForward	String	No	No	Specifies whether to enable HTTP-to-HTTPS redirection.	Valid values: • on • off Default value: off.

Response parameters

Fn::GetAtt

- LoadBalancerId: the unique ID of the SLB instance.
- ListenerPortsAndProtocol: an array consisting of the ports and protocols used by the SLB listener.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "LoadBalancer": {
      "Type": "ALIYUN::SLB::LoadBalancer",
      "Properties": {
        "LoadBalancerName": "createdByHeat",
        "AddressType": "internet",
        "InternetChargeType": "paybybandwidth"
      }
    },
    "CreateListener": {
      "Type": "ALIYUN::SLB::Listener",
      "Properties": {
        "LoadBalancerId": {"Ref": "LoadBalancer"},
        "ListenerPort": "8094",
        "BackendServerPort": 8080,
        "Bandwidth": 1,
        "Protocol": "http",
        "HealthCheck": {
          "HealthyThreshold": 3,
          "UnhealthyThreshold": 3,
          "Interval": 2,
          "Timeout": 5,
          "HttpCode": "http_2xx,http_3xx,http_4xx,http_5xx"
        },
        "Scheduler": "wrr",
          "Persistence": {
          "PersistenceTimeout": 1,
          "XForwardedFor": "on",
          "StickySession": "on",
          "StickySessionType": "insert",
          "CookieTimeout": 10,
          "Cookie": "1"
        }
      }
    }
  },
  "Outputs": {
  "LoadBalanceDetails": {
     "Value": {"Fn::GetAtt": ["LoadBalancer", "LoadBalancerId"]}
  }
  }
}
```

5.1.5.6.7. ALIYUN::SLB::LoadBalancer

ALIYUN::SLB::LoadBalancer is used to create an SLB instance.

Statement

```
{
  "Type": "ALIYUN::SLB::LoadBalancer",
  "Properties": {
    "DeletionProtection": Boolean,
    "AddressType": String,
    "Tags": List,
    "InternetChargeType": String,
    "Bandwidth": Integer,
    "SlaveZoneId": String,
    "ResourceGroupId": String,
    "AutoPay": Boolean,
    "VpcId": String,
    "PricingCycle": String,
    "LoadBalancerName": String,
    "Duration": Number,
    "VSwitchId": String,
    "LoadBalancerSpec": String,
    "MasterZoneId": String,
    "PayType": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the RDS instance belongs.	None
DeletionProtection	Boolean	Erased	Released	Specifies whether to enable deletion protection to prevent the SLB instance from being deleted by mistake.	Valid values: • true • false
VpcId	String	Yes	Released	The ID of the VPC.	None
SlaveZoneId	String	Yes	Released	The ID of the secondary zone to which the SLB instance belongs.	None

Parameter	Type	Required	Editable	Description	Constraint
Bandwidth	String	Optional	Released	The peak bandwidth of SLB instances that are connected to the Internet and billed by fixed bandwidth.	For SLB instances that are connected to the Internet and billed by fixed Bandwidth, this parameter is valid only when the Bandwidth parameter of the SLB Listener is specified. For Internet instances whose billing type is to pay by traffic, we recommend that you set the peak Bandwidth through the Listener parameter. In this case, this parameter is ignored. Valid values: 1 to 1000. Unit: Mbps. Default value: 1 VPC-type instances are billed by traffic.
AddressType	String	Yes	Released	The address type of the SLB instance.	Valid values: • internet • intranet Default value: internet.
VSwitchId	String	Yes	Released	The ID of the vSwitch in the VPC.	None

Parameter	Type	Required	Editable	Description	Constraint
LoadBalancerName	String	Yes	Released	The name of the SLB instance to be created.	A custom string. The name must be 1 to 80 characters in length and can contain letters, digits, hyphens (-), forward slashes (/), periods (.), and underscores (_). If this parameter is not specified, the system assigns a value.
InternetChargeType	String	Yes	Released	The billing method of SLB instances that are connected to the Internet.	Valid values: • paybybandwidth • paybytraffic Default value: paybytraffic.
MasterZoneId	String	Yes	Released	The ID of the primary zone to which the SLB instance belongs.	None
Tags	List	Erased	Released	The tags to be attached to the SLB instance. The tags are listed in JSON format. Each tag consists of a TagKey and a TagValue.	A maximum of five tags can be attached to an SLB instance.
LoadBalancerSpec	String	Yes	Released	The type of the SLB instance.	Valid values: • slb.s1.small • slb.s2.small • slb.s2.medium • slb.s3.small • slb.s3.medium • slb.s3.large The available types vary by region.

Parameter	Type	Required	Editable	Description	Constraint
AutoPay	Boolean	Erased	Released	Specifies whether to automatically pay for subscription SLB instances that are connected to the Internet.	Valid values: • true • false Default value: false. **Note** This parameter is valid only on the China site (aliyun.com).
PayType	String	Yes	Released	The billing method of the SLB instance.	Valid values: • PayOnDemand • PrePay
PricingCycle	String	Yes	Released	The billing cycle of subscription SLB instances that are connected to the Internet.	Valid values: • month • year **Note** This parameter is valid only on the China site (aliyun.com).

Parameter	Type	Required	Editable	Description	Constraint
Duration	Number	Erased	Released	The subscription period of subscription SLB instances that are connected to the Internet.	Valid values: • Valid values when the PricingCycle parameter is set to month: 1 to 9. • Valid values when the PricingCycle parameter is set to year: 1 to 3. **Note** This parameter is valid only on the China site (aliyun.com).

Tags syntax

```
"Tags": [
  {
    "Value": String ,
    "Key": String
  }
]
```

Tags properties

Property	Type	Required or Not	Editable	Description	Constraint
Key	String	No	No	None	None
Value	String	Yes	Released	None	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- LoadBalancerId: the unique ID of the SLB instance.

- NetworkType: the network type of the SLB instance, which can be vpc or classic.
- AddressType: the address type of the SLB instance, which can be intranet or internet.
- IpAddress: the IP address of the SLB instance.
- OrderId: the ID of the order.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "CreateLoadBalance": {
      "Type": "ALIYUN::SLB::LoadBalancer",
      "Properties": {
        "LoadBalancerName": "createdByHeat",
        "AddressType": "internet",
        "InternetChargeType": "paybybandwidth",
      }
    }
  },
  "Outputs": {
    "LoadBalanceDetails": {
      "Value": {
        "Fn::GetAtt": ["CreateLoadBalance", "LoadBalancerId"]}
    }
  }
}
```

5.1.5.6.8. ALIYUN::SLB::LoadBalancerClone

ALIYUN::SLB::LoadBalancerClone is used to clone an SLB instance.

Syntax

```
{
  "Type": "ALIYUN::SLB::LoadBalancerClone",
  "Properties": {
    "Tags": List,
    "ResourceGroupId": String,
    "VSwitchId": String,
    "LoadBalancerName": String,
    "SourceLoadBalancerId": String,
    "TagsPolicy": String,
    "BackendServersPolicy": String,
    "BackendServers": List
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	No	No	The ID of the resource group.	None

Property	Type	Required	Editable	Description	Constraint
VSwitchId	String	No	No	The ID of the VSwitch.	The VSwitch must exist in the VPC to which the source SLB instance belongs. If the parameter is not specified, the VSwitch to which the source SLB instance belongs is used.
SourceLoadBalancerId	String	Yes	No	The ID of the SLB instance to be cloned.	None

Property	Type	Required	Editable	Description	Constraint
BackendServersPolicy	String	No	No	The clone policy. The ECS instances listened by the new SLB instance and the weight of each ECS instance are specified in the policy.	Valid values: • clone: The ECS instances listened by the source SLB instance and the ECS instance weights are cloned to the new SLB instance. • empty: No ECS instances are attached to the new SLB instance. • append: The ECS instances listened by the source SLB instance and the ECS instance weights are cloned to the new SLB instance. New ECS instances with specified weights are also attached to the new SLB instance. • replace: New ECS instances with specified weights are attached to the new SLB instance. However, the ECS instances listened by the source SLB instance and the ECS instance weights are not cloned to the new SLB instance. Default value: clone.

Property	Type	Required	Editable	Description	Constraint
BackendServers	List	No	Yes	The new ECS instances to be listened.	None
LoadBalancerName	String	No	No	The name of the new SLB instance.	You can set the name to any string. The name must be 1 to 80 characters in length and can contain letters, digits, hyphens (-), forward slashes (/), periods (.), and underscores (_).
Tags	List	No	Yes	The tags of the SLB instance.	Tags must be specified as key-value pairs. A maximum of five tags can be specified.
TagsPolicy	String	No	No	The policy of the tags.	Valid values: • clone: The tags of the source SLB instance are used. • empty: No tags are configured. • append: The tags of the source SLB instance are reserved while new tags are added. • replace: The tags of the source SLB instance are deleted while new tags are added. Default value: empty.

BackendServers syntax

```
"BackendServers": [
  {
    "ServerId": String,
    "Weight": Integer
  }
]
```

BackendServers properties

Property	Type	Required	Editable	Description	Constraint
ServerId	String	Yes	Yes	The ID of the ECS instance.	The ECS instances must be in the running state.
Weight	Integer	Yes	Yes	The weight of the ECS instance to be attached to the SLB instance.	Valid values: 0 to 100. Default value: 100.

Response parameters

Fn::GetAtt

LoadBalancerId: the ID of the new SLB instance.

Examples

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Resources" : {
    "CloneLoadBalance": {
      "Type": "ALIYUN::SLB::LoadBalancerClone",
      "Properties": {
        "SourceLoadBalancerId": "150ebed5f06-cn-beijing-btc-***",
        "LoadBalancerName": "rosnew",
        "BackendServersPolicy": "replace",
        "BackendServers": [
            {
                "ServerId": "i-25zsk****",
                "Weight": 20
            }
        ]
      }
    }
  },
  "Outputs": {
    "LoadBalanceDetails": {
         "Value" : {"Fn::GetAtt": ["CloneLoadBalance", "LoadBalancerId"]}
    }
  }
}
```

5.1.5.6.9. ALIYUN::SLB::MasterSlaveServerGroup

ALIYUN::SLB::MasterSlaveServerGroup is used to create a primary/secondary server group.

Notice A primary/secondary server group contains only two ECS instances: a primary server and a secondary server.

Syntax

```
{
  "Type": "ALIYUN::SLB::MasterSlaveServerGroup",
  "Properties": {
    "MasterSlaveServerGroupName": String,
    "MasterSlaveBackendServers": List,
    "LoadBalancerId": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
MasterSlaveServerGroupName	String	No	No	The name of the primary/secondary server group.	None
MasterSlaveBackendServers	List	Yes	No	The list of backend servers in the primary/secondary server group.	A primary/secondary server group can contain a maximum of two backend servers. If you do not specify this parameter, an empty primary/secondary server group is created.
LoadBalancerId	String	Yes	No	The ID of the SLB instance.	None

MasterSlaveBackendServers syntax

```
"MasterSlaveBackendServers": [
  {
    "ServerId": String,
    "Port": Integer,
    "Weight": Integer,
    "ServerType": String
  }
]
```

MasterSlaveBackendServers properties

Property	Type	Required	Editable	Description	Constraint
ServerId	String	Yes	No	The ID of the ECS instance or Elastic Network Interface (ENI) to be added.	None
ServerType	String	No	No	The type of the server.	Default value: Master. Valid values: • Master • Slave
Port	Integer	Yes	No	The port number used by the backend server.	Valid values: 1 to 65535.
Weight	Integer	Yes	No	The weight of the backend server.	Valid values: 0 to 100.

Response parameters

Fn::GetAtt

MasterSlaveServerGroupId: the ID of the primary/secondary server group.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "MasterSlaveServerGroup": {
      "Type": "ALIYUN::SLB::MasterSlaveServerGroup",
      "Properties": {
        "MasterSlaveServerGroupName": "Group1",
        "MasterSlaveBackendServers": [
          {
            "ServerId": "vm****",
            "Port": "80",
            "Weight": "100",
            "ServerType": "Master"
          },
          {
            "ServerId": "vm****",
            "Port": "90",
            "Weight": "100",
            "ServerType": "Slave"
          }
        ],
        "LoadBalancerId": "lb-bp1hv944r69al4j9j****"
      }
    }
  },
  "Outputs": {
    "MasterSlaveServerGroupId": {
      "Value": {
        "Fn::GetAtt": [
          "MasterSlaveServerGroup",
          "MasterSlaveServerGroupId"
        ]
      }
    }
  }
}
```

5.1.5.6.10. ALIYUN::SLB::Rule

ALIYUN::SLB::Rule is used to add forwarding rules for a specified HTTP or HTTPS listener.

Statement

```
{
  "Type": "ALIYUN::SLB::Rule",
  "Properties": {
    "ListenerPort": Integer,
    "RuleList": List,
    "LoadBalancerId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ListenerPort	Integer	Yes	No	The frontend listener port used by the SLB instance.	Valid values: 1 to 65,535.
RuleList	List	Yes	No	The list of forwarding rules to be added.	A maximum of 10 forwarding rules can be added at a time. Each forwarding rule contains the following parameters: RuleName, Domain, Url, and VServerGroupId. You must specify at least one of the following parameters: Domain and URL. The combination of Domain and URL must be unique in a listener.
LoadBalancerId	String	No	No	The IDs of SLB instances.	None

RuleList syntax

```
"RuleList": [
  {
    "Url": String,
    "Domain": String,
    "VServerGroupId": String,
    "RuleName": String
  }
]
```

RuleList properties

Parameter	Type	Required	Editable	Description	Constraint

Parameter	Type	Required	Editable	Description	Constraint
Url	String	Yes	Released	The request URL.	The name must be 1 to 80 characters in length and can contain letters, numbers, and special characters. `-/.` percent signs (%), question marks (?), #& .
Domain	String	Yes	Released	The request domain name associated with the forwarding rule.	None
VServerGroupId	String	No	No	The ID of the destination VServer group specified in the forwarding rule.	None
RuleName	String	No	No	The name of the forwarding rule.	The name must be 1 to 40 characters in length and can contain letters, numbers, and special characters. `-/._` . Forwarding rule names must be unique within each listener.

Response parameters

Fn::GetAtt

Rules: the list of forwarding rules.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Rule": {
      "Type": "ALIYUN::SLB::Rule",
      "Properties": {
        "ListenerPort": {
          "Ref": "ListenerPort"
        },
        "RuleList": {
          "Fn::Split": [",", {
            "Ref": "RuleList"
          }, {
            "Ref": "RuleList"
          }]
        },
        "LoadBalancerId": {
          "Ref": "LoadBalancerId"
        }
      }
    }
  },
  "Parameters": {
    "ListenerPort": {
      "Type": "Number",
      "Description": "The front-end HTTPS listener port of the Server Load Balancer instance. Valid v
alue:\n1-65535",
      "MaxValue": 65535,
      "MinValue": 1
    },
    "RuleList": {
      "MinLength": 1,
      "Type": "CommaDelimitedList",
      "Description": "The forwarding rules to add.",
      "MaxLength": 10
    },
    "LoadBalancerId": {
      "Type": "String",
      "Description": "The ID of Server Load Balancer instance."
    }
  },
  "Outputs": {
    "Rules": {
      "Description": "A list of forwarding rules. Each element of rules contains \"RuleId\".",
      "Value": {
        "Fn::GetAtt": ["Rule", "Rules"]
      }
    }
  }
}
```

5.1.5.6.11. ALIYUN::SLB::VServerGroup

ALIYUN::SLB::VServerGroup is used to create a VServer group and adds backend servers to the SLB instance.

Syntax

```
{
    "Type" : "ALIYUN::SLB::VServerGroup",
    "Properties" : {
         "VServerGroupName" : String,
         "BackendServers" : List,
         "LoadBalancerId" : String
    }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
VServerGroupName	String	Yes	No	The name of the VServer group.	None
BackendServers	List	Yes	Yes	The list of ECS instances to be added.	A list can contain up to 20 instances.
LoadBalancerId	String	Yes	No	The ID of the SLB instance.	None

BackendServers syntax

```
"BackendServers" : [
    {
        "ServerId" : String,
        "Port" : Integer,
        "Weight" : Integer
    }
]
```

BackendServers properties

Property	Type	Required	Editable	Description	Constraint
ServerId	String	Yes	Yes	The ID of the ECS instance.	None
Port	Integer	Yes	Yes	The backend port used by the SLB instance.	Valid values: 1 to 65535.
Weight	Integer	Yes	Yes	The weight of the ECS instance to be attached to the SLB instance.	Valid values: 0 to 100.

Response parameters

Fn::GetAtt

- VServerGroupId: the ID of the VServer group.

- BackendServers: the list of backend servers added to the SLB instance

Examples

```
{
    "ROSTemplateFormatVersion": "2015-09-01",
    "Resources": {
        "CreateVServerGroup": {
            "Type": "ALIYUN::SLB::VServerGroup",
            "Properties": {
                "LoadBalancerId": "lb-2zenh4ndwrqg14yt0****",
                "VServerGroupName": "VServerGroup-****",
                "BackendServers": [
                    {
                        "ServerId": "i-25zsk****",
                        "Weight": 20,
                        "Port": 8080
                    },
                    {
                        "ServerId": "i-25zsk****",
                        "Weight": 100,
                        "Port": 8081
                    }
                ]
            }
        }
    },
    "Outputs": {
        "VServerGroupId": {
            "Value" : {"Fn::GetAttr": ["CreateVServerGroup",  "VServerGroupId"]
         }
      }
   }
}
```

5.1.5.7. VPC

5.1.5.7.1. ALIYUN::VPC::EIP

ALIYUN::VPC::EIP is used to apply for an Elastic IP address.

Statement

```
{
  "Type": "ALIYUN::VPC::EIP",
  "Properties": {
    "Isp": String,
    "Period": Number,
    "ResourceGroupId": String,
    "AutoPay": Boolean,
    "InstanceChargeType": String,
    "PricingCycle": String,
    "InternetChargeType": String,
    "Bandwidth": Number
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the RDS instance belongs.	None
Bandwidth	Number	Erased	Released	The network bandwidth. Unit: Mbit/s.	If this parameter is not specified, the default value 5Mbps is used.
InternetChargeType	String	Yes	Released	The billing method for network usage. Default value: PayByBandwidth.	Valid values: • PayByBandwidth: pay-by-bandwidth. • PayByTraffic Default value: PayByBandwidth.
InstanceChargeType	String	Yes	Released	The billing method of the Elastic IP address. Default value: Postpaid.	Valid values: • Prepaid • Postpaid: pay-as-you-go Default value: PostPaid.
PricingCycle	String	Yes	Released	The billing cycle of the subscription. Default value: Month.	Valid values: • Month: paid by month. • Year Default value: Month. **Note** This parameter is required when InstanceChargeType is set to Prepaid.
Period	Number	Erased	Released	The subscription period.	Valid values: • If pay by month is selected, the billing method can be a fee of 1 to 9. • If pay-as-you-go is selected, the payment can be in the range of 1 to 3. Default value: 1 **Note** This parameter is required when InstanceChargeType is set to Prepaid.

Parameter	Type	Required	Editable	Description	Constraint
AutoPay	Boolean	Erased	Released	Specifies whether to enable automatic payment.	Valid values: • false: Automatic payment is disabled. After an order is generated, you must go to the Order Center to make the payment. • true: Automatic payment is enabled. Payments are automatically made. **Note** This parameter is required when InstanceChargeType is set to Prepaid.
Isp	String	Yes	Released	The ISP tag used for Finance Cloud. This parameter takes effect only when your region is set to China (Hangzhou).	This parameter is ignored if you are not a Finance Cloud user.

Response parameters

Fn::GetAtt

- EipAddress: the allocated Elastic IP address.
- AllocationId: the ID of the instance that the Elastic IP address is allocated to.
- OrderId: The order ID that is returned when you set the InstanceChargeType parameter to Prepaid.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Eip": {
      "Type": "ALIYUN::VPC::EIP",
      "Properties": {
        "InternetChargeType": "PayByTraffic",
          "Bandwidth": 200
      }
    }
  },
  "Outputs": {
    "EipAddress": {
      "Value" : {"Fn::GetAtt": ["Eip", "EipAddress"]}
    },
    "AllocationId": {
      "Value" : {"Fn::GetAtt": ["Eip", "AllocationId"]}
    },
    "OrderId": {
      "Value" : {"Fn::GetAtt": ["Eip", "OrderId"]}
    }
  }
}
```

5.1.5.7.2. ALIYUN::VPC::EIPAssociation

ALIYUN::VPC::EIPAssociation is used to associate an Elastic IP address with a cloud service instance.

Statement

```
{
  "Type": "ALIYUN::VPC::EIPAssociation",
  "Properties": {
    "AllocationId": String,
    "InstanceId": String,
    "PrivateIpAddress": String,
    "Mode": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
AllocationId	String	No	Yes	The ID of the Elastic IP address.	None

Parameter	Type	Required	Editable	Description	Constraint
InstanceId	String	No	Yes	The ID of the cloud service instance.	The following instance types are supported: • VPC-connected ECS instances • VPC-connected SLB instances • NAT gateways • HAVIP • Elastic network interfaces
PrivateIpAddress	String	Yes	True	The private IP address in the CIDR block of the VSwitch.	None
Mode	String	Yes	True	The association mode.	Valid values: • NAT • MULTI_BINDED

Response parameters

Fn::GetAtt

- EipAddress: The allocated Elastic IP address.
- AllocationId: The ID of the instance to which the Elastic IP address is allocated.

Examples

`JSON` format

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Eip": {
      "Type": "ALIYUN::VPC::EIP",
      "Properties": {
        "InternetChargeType": "PayByTraffic",
          "Bandwidth": 200
      }
    },
    "EipAssociation": {
      "Type": "ALIYUN::VPC::EIPAssociation",
      "Properties": {
        "InstanceId": "<LoadBalancerId>",
        "InstanceType": "EcsInstance",
        "AllocationId": {
          "Fn::GetAtt": ["Eip", "AllocationId"]
        }
      }
    }
  },
  "Outputs": {
    "EipAddress": {
      "Value" : {"Fn::GetAtt": ["EipAssociation", "EipAddress"]}
    },
    "AllocationId": {
      "Value" : {"Fn::GetAtt": ["EipAssociation", "AllocationId"]}
    }
  }
}
```

`YAML` format

```
ROSTemplateFormatVersion: '2015-09-01'
Resources:
  Eip:
    Type: ALIYUN::VPC::EIP
    Properties:
      InternetChargeType: PayByTraffic
      Bandwidth: 200
  EipAssociation:
    Type: ALIYUN::VPC::EIPAssociation
    Properties:
      InstanceId: "<LoadBalancerId>"
      InstanceType: EcsInstance
      AllocationId:
        Fn::GetAtt:
        - Eip
        - AllocationId
Outputs:
  EipAddress:
    Value:
      Fn::GetAtt:
      - EipAssociation
      - EipAddress
  AllocationId:
    Value:
      Fn::GetAtt:
      - EipAssociation
      - AllocationId
```

5.1.5.7.3. ALIYUN::VPC::PeeringRouterInterfaceBinding

ALIYUN::VPC::PeeringRouterInterfaceBinding is used to associate two router interfaces to be interconnected.

Statement

```
{
  "Type": "ALIYUN::VPC::PeeringRouterInterfaceBinding",
  "Properties": {
    "OppositeRouterId": String,
    "OppositeInterfaceId": String,
    "OppositeInterfaceOwnerId": String,
    "RouterInterfaceId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
RouterInterfaceId	String	No	No	The ID of the router interface.	None
OppositeInterfaceId	String	No	No	The ID of the peer router interface.	None

Parameter	Type	Required	Editable	Description	Constraint
OppositeRouterId	String	Yes	Released	The ID of the router to which the peer router interface belongs.	None
OppositeInterfaceOwnerId	String	Yes	Released	The ID of the owner of the peer router interface.	None

Response parameters

Fn::GetAtt

RouterInterfaceId: the ID of the vRouter.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "InitiatorRouterInterfaceBinding": {
      "Type": "ALIYUN::VPC::PeeringRouterInterfaceBinding",
      "Properties": {
        "RouterInterfaceId": "ri-2zedgo0ih64g1me29****",
        "OppositeInterfaceId": "ri-2zex1tkyym98pjaor****",
        "OppositeRouterId": "vrt-2zexb35tzoriu0286****"
      }
    }
  }
}
```

5.1.5.7.4. ALIYUN::VPC::PeeringRouterInterfaceConnection

ALIYUN::VPC::PeeringRouterInterfaceConnection is used to initiate a router interface connection.

Statement

```
{
  "Type": "ALIYUN::VPC::PeeringRouterInterfaceConnection",
  "Properties": {
    "OppositeInterfaceId": String,
    "RouterInterfaceId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
OppositeInterfaceId	String	No	No	The ID of the acceptor router interface.	None

Parameter	Type	Required	Editable	Description	Constraint
RouterInterfaceId	String	No	No	The ID of the router interface to initiate the connection.	None

Response parameters

Fn::GetAtt

- OppositeInterfaceId: the ID of the acceptor router interface.
- RouterInterfaceId: the ID of the router interface that initiates the connection.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "InitiatorRouterInterfaceBinding": {
      "Type": "ALIYUN::VPC::PeeringRouterInterfaceConnection",
      "Properties": {
        "RouterInterfaceId": "ri-2zedgo0ih64g1me29****",
        "OppositeInterfaceId": "ri-2ze4k5n2aeardu8cy****"
      }
    }
  }
}
```

5.1.5.7.5. ALIYUN::VPC::RouterInterface

ALIYUN::VPC::RouterInterface is used to create a router interface.

Syntax

```
{
  "Type": "ALIYUN::VPC::RouterInterface",
  "Properties": {
    "OppositeRegionId": String,
    "Description": String,
    "HealthCheckSourceIp": String,
    "RouterType": String,
    "AccessPointId": String,
    "RouterId": String,
    "Role": String,
    "OppositeInterfaceOwnerId": String,
    "OppositeAccessPointId": String,
    "HealthCheckTargetIp": String,
    "OppositeRouterId": String,
    "Spec": String,
    "OppositeRouterType": String,
    "Name": String,
    "PricingCycle": String,
    "Period": Number,
    "AutoPay": Boolean,
    "InstanceChargeType": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
RouterId	String	Yes	No	The ID of the router	None
Role	String	Yes	No	The role of the router interface.	• When RouterType is set to VBR, set the value to InitiatingSide. • When OppositeRouterType is set to VBR, set the value to AcceptingSide.
RouterType	String	No	No	The type of the router to which the router interface belongs.	Valid values: • VRouter • VBR

Property	Type	Required	Editable	Description	Constraint
AccessPointId	String	No	No	The ID of the access point of the router interface.	• This parameter is required when RouterType is set to VBR. The access point ID cannot be modified after the router interface is created. • This parameter is not required when RouterType is set to VRouter.

Property	Type	Required	Editable	Description	Constraint
Spec	String	No	No	The specifications of the router interface.	The following list includes available specifications and the corresponding bandwidth values: • Mini.2: 2 Mbit/s • Mini.5: 5 Mbit/s • Small.1: 10 Mbit/s • Small.2: 20 Mbit/s • Small.5: 50 Mbit/s • Middle.1: 100 Mbit/s • Middle.2: 200 Mbit/s • Middle.5: 500 Mbit/s • Large.1: 1,000 Mbit/s • Large.2: 2,000 Mbit/s • Large.5: 5,000 Mbit/s • Xlarge.1: 10,000 Mbit/s Note • This parameter is required when Role is set to InitiatingSide. • The value Negative is used by default when Role is set to AcceptingSide.
OppositeRegionId	String	No	No	The region ID of the peer router interface.	None
OppositeInterfaceOwnerId	String	No	No	The ID of the owner of the peer router interface.	The default value is the ID of the current user.

Property	Type	Required	Editable	Description	Constraint
OppositeRouterId	String	No	No	The ID of the router to which the peer router interface belongs.	None
OppositeRouterType	String	No	No	The type of the router to which the peer router interface belongs.	Valid values: • When RouterType is set to VBR, set the value to VRouter. • VBR
OppositeAccessPointId	String	No	No	The ID of the access point of the peer router interface.	• When OppositeRouterType is set to VBR, this parameter is required . The access point ID cannot be modified after the router interface is created. • When OppositeRouterType is set to VRouter, this parameter is not required. • When OppositeRouterType is set to VBR, the VBR specified by the OppositeRouterId parameter must be in the access point specified by the OppositeAccessPointId parameter.
Description	String	No	No	The description of the router interface.	The description must be 2 to 256 characters in length. It cannot start with `http://` or `https://` . The parameter is empty by default.

Property	Type	Required	Editable	Description	Constraint
Name	String	No	No	The display name of the router interface.	• The name must be 2 to 128 characters in length and can contain letters, digits, periods(.), underscores (_), and hyphens (-). • It must start with a letter but cannot start with `http://` or `https://` .
HealthCheckSourceIp	String	No	No	The source IP address of health check packets used in leased line disaster recovery and ECMP scenarios.	This parameter is valid only for VRouter interfaces with a peer router interface on a VBR. It must be an unused IP address in the VPC where the local VRouter is located. The HealthCheckSourceIp and HealthCheckTargetIp parameters must either both be specified or both left unspecified.

Property	Type	Required	Editable	Description	Constraint
HealthCheckTargetIp	String	No	No	The destination IP address of health check packets used in leased line disaster recovery and ECMP scenarios.	This parameter is valid only for VRouter interfaces with a peer router interface on a VBR. Typically, you can use the IP address of a customer premises equipment (CPE) on the user side of the leased line, which is the IP address of the peer gateway on the VBR where the peer router interface is located. You can also specify another IP address on the user side of the leased line as the destination IP address. The HealthCheckSourceIp and HealthCheckTargetIp parameters must either both be specified or both left unspecified.
PricingCycle	String	No	No	The billing cycle of the subscription.	Valid values: • Month • Year
Period	Number	No	No	The subscription duration.	• Valid values when the PricingCycle parameter is set to Month: 1 to 9. • Valid values when the PricingCycle parameter is set to Year: 1 to 3.
AutoPay	Boolean	No	No	Specifies whether to enable automatic payment.	Default value: false. Valid values: • true • false

Property	Type	Required	Editable	Description	Constraint
InstanceChargeType	String	No	No	The billing method of the instance.	Valid values: • Postpaid: pay-as-you-go • Prepaid: subscription

Response parameters

Fn::GetAtt

RouterInterfaceId: the ID of the router interface.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "RouterInterface": {
      "Type": "ALIYUN::VPC::RouterInterface",
      "Properties": {
        "Name": "RouterInterface_1",
        "Description": "VPC initiator RouterInterface",
        "RouterId": "vrt-2ze2i147e5n0bicoe****",
        "Role": "AcceptingSide",
        "OppositeRegionId": "cn-beijing",
        "HealthCheckSourceIp": "10.0.XX.XX",
        "HealthCheckTargetIp": "192.168.XX.XX"
      }
    }
  },
  "Outputs": {
    "RouterInterfaceId": {
      "Value": {"Fn::GetAtt": ["RouterInterface", "RouterInterfaceId"]}
    }
  }
}
```

6.Object Storage Service (OSS)

6.1. User Guide

6.1.1. What is OSS?

Object Storage Service (OSS) is a secure, cost-effective, and highly reliable cloud storage service provided by Alibaba Cloud. It enables you to store a large amount of data in the cloud.

Compared with user-created server storage, OSS has outstanding advantages in reliability, security, cost-effectiveness, and data processing capabilities. OSS enables you to store and retrieve a variety of unstructured data objects, such as text, images, audios, and videos over the network at any time.

OSS is an object storage service based on key-value pairs. Files uploaded to OSS are stored as objects in buckets. You can obtain the content of an object based on the object key.

In OSS, you can perform the following operations:

- Create a bucket and upload objects to the bucket.
- Obtain an object URL from OSS to share or download the object.
- Modify the attributes or metadata of a bucket or an object. You can also configure the ACL of the bucket or the object.
- Perform basic and advanced operations in the OSS console.
- Perform basic and advanced operations by using OSS SDKs or calling RESTful API operations in your application.

6.1.2. Usage notes

Before you use OSS, you must understand the following content:

To allow other users to use all or part of OSS features, you must create RAM users and grant permissions to the users by configuring RAM policies.

Before you use OSS, you must also understand the following limits.

Item	Limit
Bucket	• You can create up to 100 buckets. • After a bucket is created, its name and region cannot be modified.
Upload objects	• Objects larger than 5 GB cannot be uploaded by using the following modes: console upload, simple upload, form upload, or append upload. To upload an object that is larger than 5 GB, you must use multipart upload. The size of an object uploaded by using multipart upload cannot exceed 48.8 TB. • If you upload an object that has the same name of an existing object in OSS, the new object will overwrite the existing object.
Delete objects	• Deleted objects cannot be recovered. • You can delete up to 100 objects at a time in the OSS console. To delete more than 100 objects at a time, you must call an API operation or use an SDK.
Lifecycle	You can configure up to 1,000 lifecycle rules for each bucket.

6.1.3. Quick start

6.1.3.1. Log on to the OSS console

This topic describes how to log on to the OSS console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use Google Chrome.

Procedure

1. In the address bar, enter the URL used to access the Apsara Uni-manager Management Console. Press Enter.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Login** to go to the Apsara Uni-manager Management Console.
4. In the top navigation bar, choose **Products > Object Storage Service**.

6.1.3.2. Create buckets

Objects uploaded to OSS are stored in a bucket. You must create a bucket before you upload objects to OSS.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click **Create Bucket**.
3. On the **Create OSS Bucket** page, configure parameters.

   The following table describes the parameters that you can configure.

Parameter	Description
Organization	Select an organization from the drop-down list for the bucket.
Resource Set	Select a resource set from the drop-down list for the bucket.

Parameter	Description
Region	Select a region from the drop-down list for the bucket. **Note** ◦ The region of a bucket cannot be changed after the bucket is created. ◦ If you want to access OSS from your ECS instance through the internal network, select the same region where your ECS instance is deployed.
Cluster	Select a cluster for the bucket.
Bucket Name	Enter the name of the bucket. **Note** ◦ The bucket name must comply with the naming conventions. ◦ The bucket name must be globally unique among all existing buckets in OSS. ◦ The bucket name cannot be changed after the bucket is created.
Storage Class	Set the value to **Standard**. Only Standard is supported.
Bucket Capacity	Specify the capacity of the bucket. Valid values: 0 to 2000000. Unit: TB or GB.
Access Control List (ACL)	Set the ACL of the bucket. You can select the following options: ◦ **Private**: Only the owner or authorized users of this bucket can read and write objects in the bucket. Other users, including anonymous users cannot access objects in the bucket without authorization. ◦ **Public Read**: Only the owner or authorized users of this bucket can read and write objects in the bucket. Other users, including anonymous users can only read objects in the bucket. ◦ **Public Read/Write**: All users, including anonymous users can read and write objects in the bucket. Fees incurred by such operations are paid by the owner of the bucket. Exercise caution when you configure this option. **Note** You can modify the ACL of a bucket after the bucket is created. For more information, see Modify bucket ACLs.

Parameter	Description
Server-Side Encryption	Configure server-side encryption for the bucket. You can select the following options: ○ **None**: Server-side encryption is not performed. ○ **AES256**: AES256 is used to encrypt each object in the bucket using different data keys. Customer master keys (CMKs) used to encrypt the data keys are rotated regularly. ○ **KMS**: CMKs managed by KMS are sued to encrypt objects in the bucket.
Encryption Algorithm	You can configure this parameter when you select **KMS** for **Server-Side Encryption**.
Key ID	You can configure this parameter when you select **KMS** for **Server-Side encryption**. OSS uses the specified CMK to encrypt objects in the bucket. **Note** To select a CMK ID for server-side encryption, you must create the CMK in the KMS console.

4. Click **Submit**.

6.1.3.3. Upload objects

After you create a bucket, you can upload objects to it.

Prerequisites

A bucket is created. For more information about how to create a bucket, see Create buckets.

Context

You can upload an object of any format to a bucket. You can use the OSS console to upload an object up to 5 GB in size. To upload an object larger than 5 GB, use OSS SDKs or call an API operation.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket to which you want to upload objects.
3. On the bucket details page, click the **Files** tab.
4. Click **Upload**.
5. In the **Upload** panel, set the parameters described in the following table.

Parameter	Description
Upload To	Set the folder to which objects are uploaded. ○ **Current**: Objects are uploaded to the current folder. ○ **Specified**: Objects are uploaded to the specified folder. You must enter a folder name. If the specified folder does not exist, OSS automatically creates the specified folder and uploads the object to the folder.

Parameter	Description
File ACL	Set the ACL of the object to upload. Default value: **Inherited from Bucket**. ○ **Inherited from Bucket**: The ACL of uploaded objects is the same as that of the bucket. ○ **Private**: Only the owner or authorized users can read and write objects in the bucket. Other users, including anonymous users cannot access the objects in the bucket without authorization. ○ **Public Read**: Only the bucket owner can perform write operations on objects in the bucket. Other users, including anonymous users, can perform only read operations on objects in the bucket. ○ **Public Read/Write**: All users, including anonymous users, can read and write objects in the bucket.
Upload	Drag one or more objects to upload to this section, or click **Upload** to select one or more objects to upload. **Notice** ○ When the uploaded object has the same name as an existing object in the bucket, the existing object is overwritten. ○ If you upload a folder, only the files in the folder are uploaded and stored in the same folder in the bucket. ○ The name of an uploaded object must comply with the following conventions: ■ The name can contain only UTF-8 characters. ■ The name is case-sensitive. ■ The name must be 1 to 1,023 bytes in length. ■ The name cannot start with a forward slash (/) or backslash (\).

6. In the **Upload Tasks** panel, wait until the upload task is complete.

   Do not refresh or close the **Upload Tasks** panel when objects are being uploaded. Otherwise, the upload tasks are interrupted.

6.1.3.4. Obtain object URLs

You can obtain the URL of an uploaded object and share the URL with other users to preview or download the object.

Prerequisites

An object is uploaded to the bucket. For more information, see Upload objects.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket in which the object whose URL you want to obtain is stored.
3. Click the **Files** tab.
4. Click the name of the object whose URL you want to obtain. In the **View Details** panel, click **Copy File URL**.

6.1.4. Buckets

6.1.4.1. View bucket information

You can view the detailed information about the created buckets in the OSS console.

Prerequisites

A bucket is created. For more information about how to create a bucket, see Create buckets.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket that you want to view.
3. On the **Overview** tab, you can view the information about the bucket, including the organization, resource set, endpoints, and basic settings.

6.1.4.2. Delete a bucket

You can delete a bucket in the OSS console.

Prerequisites

All objects and parts in the bucket are deleted. For more information, see Delete objects and Manage parts.

Warning Deleted objects, parts, and buckets cannot be recovered. Exercise caution when you delete objects, parts, and buckets.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket that you want to delete.
3. On the bucket details page that appears, click **Delete Bucket** in the upper right corner. In the message that appears, click **OK**.

6.1.4.3. Modify bucket ACLs

OSS provides access control list (ACL) to control access to buckets. By default, the ACL of bucket is private. You can modify the ACL of the bucket after it is created.

Prerequisites

A bucket is created. For more information about how to create a bucket, see Create buckets.

Context

You can set the ACL of a bucket to one of the following values:

- Private: Only the owner or authorized users of this bucket can read and write objects in the bucket. Other users, including anonymous users cannot access the objects in the bucket without authorization.
- Public Read: Only the owner or authorized users of this bucket can write objects in the bucket. Other users, including anonymous users can only read objects in the bucket.
- Public Read/Write: Any users, including anonymous users can read and write objects in the bucket.

Warning If you set the ACL of a bucket to Public Read or Public Read/Write, other users can read the data in the bucket without authentication, which results in security risks. To ensure the security of your data, we recommend that you configure the ACL of your bucket to private.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to modify the ACL.
3. Click the **Basic Settings** tab. Find the **Access Control List (ACL)** section.
4. Click **Configure**. Modify the bucket ACL.
5. Click **Save**.

6.1.4.4. Configure static website hosting

You can configure static website hosting for a bucket in the OSS console so that users can access the website by using the domain name of the bucket.

Prerequisites

A bucket is created. For more information about how to create a bucket, see Create buckets.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure static website hosting.
3. Click the **Basic Settings** tab. Find the **Static Pages** section.
4. Click **Configure** and then set the parameters described in the following table.

Parameter	Description
Default Homepage	Specify an index page that functions similar to index.html. Only HTML objects can be set to the index page. Static website hosting is disabled if you do not specify this parameter. ◦ If Subfolder Homepage is disabled, you must make sure that the index object exists in the root folder and is readable. ◦ If Subfolder Homepage is enabled, you must make sure that the index object exists in both the root folder and the subfolder and the index object is readable. **Note** You can specify only one index object for Default Homepage. If you enable Subfolder Homepage, the index object for the subfolder homepage must have the same name as that for the root folder homepage. However, the content of the index objects can be different.
Default 404 Page	Set the default 404 page that is displayed when the requested resource does not exist. Only the HTML, JPG, PNG, BMP, or WebP object in the root folder can be set to the default 404 page. Default 404 Page is disabled if you do not specify this parameter.

Parameter	Description
Subfolder Homepage	Specify whether to enable the subfolder homepage feature. ○ **Disable**: disables Subfolder Homepage. The default homepage of the root folder is displayed if you access the root domain name of a static website or any URL that ends with a forward slash (/) under this domain name. ○ **Enable**: enables Subfolder Homepage. When you access the root domain name of a static website, the default homepage of the root folder is displayed. When you access a URL ending with a forward slash (/), the default homepage of the corresponding folder is displayed. For example, when you access the URL test.oss-cn-hangzhou.aliyuncs.com/subdir/, the default homepage of the subfolder is displayed if the index object exists in the *subdir* /folder.
Subfolder 404 Rule	This parameter is available if you enable Subfolder Homepage. You can configure this parameter to specify the result to return when you access an object whose name does not end with a forward slash (/) and the object does not exist. For example, if *subdir* does not exist when you access test.oss-cn-hangzhou.aliyuncs.com/subdir, the following rules apply: ○ **Redirect**: the default rule that checks whether *subdir/index object* exists. ■ If the index object exists, HTTP status code 302 is returned with the Location header that specifies `test.oss-cn-hangzhou.aliyuncs.com/subdir/` . ■ If the index object does not exist, the default 404 page is returned. If the default 404 page does not exist, HTTP status code 404 is returned. ○ **NoSuckKey**: returns the default 404 page. If the default 404 page does not exist, HTTP status code 404 is returned. ○ **Index**: the rule that checks whether *subdir/index document* exists. ■ If the index document exists, the content of the index document is directly returned. ■ If the index document does not exist, the default 404 page is returned. If the default 404 page does not exist, HTTP status code 404 is returned.

5. Click **Save**.

6.1.4.5. Configure hotlink protection

You can configure hotlink protection for a bucket in the OSS console to prevent data in your bucket from being accessed by unauthorized domain names.

Prerequisites

A bucket is created. For more information about how to create a bucket, see Create buckets.

Context

The hotlink protection feature allows you to configure a Referer whitelist for a bucket. This way, only requests from domain names included in the Referer whitelist can access data in the bucket. OSS allows you to configure Referer whitelists based on the Referer header field in HTTP and HTTPS requests.

After hotlink protection is configured for a bucket, OSS verifies requests to objects in the bucket only when the requests are initiated from signed URLs or anonymous users. Requests that contain the Authorization field in the header are not verified.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure hotlink protection.
3. Click the **Basic Settings** tab. Find the **Hotlink Protection** section.
4. Click **Configure**. Configure the parameters.
   - Enter domain names or IP addresses in the **Referer Whitelist** field. Separate multiple Referers by using line feed. You can use asterisks (*) and question marks (?) as wildcards. Example:
     - If you add www.example.com to the Referer whitelist, requests sent from URLs that start with *www.example.com*, such as *www.example.com/123* and *www.example.com.cn* are allowed.
     - If you add *www.example.com/ to the Referer whitelist, requests sent from *http://www.example.com/* and *https://www.example.com/* are allowed.
     - An asterisk (*) can be used as a wildcard to indicate zero or more characters. For example, if you add *.example.com to the Referer whitelist, requests sent from URLs such as *help.example.com* and *www.example.com* are allowed.
     - A question mark (?) can be used as a wildcard to indicate a single character. For example, if you add example?.com to the Referer whitelist, requests sent from URLs such as *examplea.com* and *exampleb.com* are allowed.
     - You can add domain names or IP addresses that include a port number, such as *www.example.com:8080* and *10.10.10.10:8080*, to the Referer whitelist.
   - Select whether to turn on **Allow Empty Referer** to allow requests in which the Referer field is empty.

     An HTTP or HTTPS request that contains an empty Referer indicates that the request does not contain the Referer field or the value of the Referer field is empty.

     If you do not allow empty Referers fields, only HTTP or HTTPS requests which include an allowed Referer field can access the objects in the bucket.

   > **Note** By default, if you use the bucket endpoint to preview an MP4 object, the browser sends a request that contains the Referer field and a request that does not contain the Referer field at the same time. Therefore, you must not only add the bucket endpoint to the Referer whitelist but also allow empty Referer fields. To use the bucket endpoint to preview an object of other formats, you need only to allow empty Referer fields.

5. Click **Save**.

6.1.4.6. Configure logging

When you access OSS, a large number of access logs are generated. You can use the logging feature to store OSS access logs in a specified bucket.

Prerequisites

A bucket is created. For more information about how to create a bucket, see Create buckets.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure logging.
3. Click the **Basic Settings** tab. Find the **Logging** section.
4. Click **Configure**. Turn on the **Logging** switch. Select **Destination Bucket** and set **Log Prefix**.

- **Destination Bucket**: Select the name of the bucket used to store access logs from the drop-down list. You must be the owner of the selected bucket, and the bucket must be in the same region as the bucket for which logging is enabled.
- **Log Prefix**: Enter the prefix and folder where the access logs are stored. If you specify *log/<TargetPrefix>* as the prefix, access logs are stored in the *log/* directory.

5. Click **Save**.

6.1.4.7. Configure CORS

You can configure cross-origin resource sharing (CORS) in the OSS console to enable cross-origin access.

Prerequisites

A bucket is created. For more information about how to create a bucket, see Create buckets.

Context

OSS provides CORS over HTML5 to implement cross-origin access. When OSS receives a cross-origin request (or an OPTIONS request) for a bucket, OSS reads the CORS rules of the bucket and checks the relevant permissions. OSS matches the request with the rules one by one. When OSS finds the first match, OSS returns a corresponding header in the response. If no match is found, OSS does not include any CORS header in the response.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure CORS.
3. Click the **Basic Settings** tab. In the **Cross-Origin Resource Sharing (CORS)** section, click **Configure**.
4. Click **Create Rule**. In the **Create Rule** panel, configure the parameters described in the following table.

Parameter	Required	Description
Sources	Yes	Specifies the sources from which you want to allow cross-origin requests. Note the following rules when you configure the sources: ○ You can configure multiple rules for sources. Separate multiple rules with line feeds. ○ The domain names must include the protocol name, such as HTTP or HTTPS. ○ Asterisks (*) are supported as wildcards. Each rule can contain up to one asterisk (*). ○ A domain name must contain the port number if the domain name does not use the default port. Example: https://www.example.com:8080. The following examples show how to configure domain names: ○ To match a specified domain name, enter the full domain name. Example: https://www.example.com:8080. ○ Use an asterisk (*) as a wildcard in the domain name to match second-level domains. Example: https://*.example.com. ○ Enter only an asterisk (*) as the wildcard to match all domain names.

Parameter	Required	Description
Allowed Methods	Yes	Specifies the cross-origin request methods that are allowed.
Allowed Headers	No	Specifies the response headers for the allowed cross-origin requests. Take note of the following rules when you configure the allowed headers: ○ This parameter is in the key:value format and case-insensitive. Example: content-type:text/plain. ○ You can configure multiple rules for allowed headers. Separate multiple rules with new lines. ○ Each rule can contain up to one asterisk (*) as the wildcard. Set this parameter to an asterisk (*) if you do not have special requirements.
Exposed Headers	No	Specifies the response headers for allowed access requests from applications, such as an XMLHttpRequest object in JavaScript. Exposed headers cannot contain asterisks (*).
Cache Timeout (Seconds)	No	Specifies the time the browser can cache the response to a preflight (OPTIONS) request to a specific resource.

Note You can configure up to 10 rules for each bucket.

5. Click **OK**.

6.1.4.8. Configure lifecycle rules

You can configure lifecycle rules for a bucket and manage the rules in the OSS console.

Prerequisites

A bucket is created. For more information, see Create buckets.

Context

Take note of the following items when you configure lifecycle rules for a bucket:

- After a lifecycle rule is configured, it is loaded within 24 hours and takes effect within 24 hours after it is loaded. Check the configurations of a rule before you save the rule.
- Objects that are deleted based on lifecycle rules cannot be recovered. Configure lifecycle rules based on your requirements.
- You can configure up to 100 lifecycle rules in the OSS console. To configure more than 100 rules, use ossutil or OSS SDKs.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure lifecycle rules.
3. Click the **Basic Settings** tab. Find the **Lifecycle** section. Click **Configure**.
4. Click **Create Rule**. In the **Create Rule** panel, configure the parameters described in the following table.

Parameter	Description
Basic Settings	
Status	Specify the status of the lifecycle rule. Valid values: **Enabled** and **Disabled**.
Applied To	Select policies used to match objects with the rule. You can select **Files with Specified Prefix** or **Whole Bucket**. Files with Specified Prefix indicates that this rule applies to objects whose names contain a specified prefix. Whole Bucket indicates that this rule applies to all objects in the bucket. **Note** If you select **Files with Specified Prefix**, you can configure multiple lifecycle rules for objects whose names contain different prefixes. If you select **Whole Bucket**, only one lifecycle rule can be configured for the bucket.
Prefix	If you set **Applied To** to **Files with Specified Prefix**, you must specify the prefix of the objects to which the rule applies. For example, if you want the rule applies to objects whose names start with img, enter *img*.
Clear Policy	
File Lifecycle	Configure rules for objects to specify when objects expire. You can set File Lifecycle to **Validity Period (Days)**, **Expiration Date**, or **Disabled**. If you select **Disabled**, the configurations of File Lifecycle do not take effect.
Delete	Specify when objects expire based on **Validity Period (Days)** or **Expiration Date** that you set for **File Lifecycle**. Expired objects are deleted. ◦ **Validity Period (Days)**: Specify the number of days to retain objects after they are last modified. The objects are deleted the next day after they expire. For example, if you set Validity Period (Days) to 30, objects that are last modified on January 1, 2019 are deleted on February 1, 2019. ◦ **Expiration Date**: Specify the expiration date. Objects that are last modified before this date expire and are deleted. For example, if you set Expiration Date to 2019-1-1, objects that are last modified before January 1, 2019 are deleted.
Delete Parts	
Part Lifecycle	Specify the operations to perform on expired parts. You can set Part Lifecycle to **Validity Period (Days)**, **Expiration Data**, or **Disabled**. If you select **Disabled**, the configurations of Part Lifecycle do not take effect. **Notice** You must configure at least one of **File Lifecycle** and **Part Lifecycle**.
Delete	Specify when parts expire based on **Validity Period (Days)** or **Expiration Data** that you set for **Part Lifecycle**. Expired parts are deleted. You can configure this parameter in the same way as you configure the Delete parameter in Clear Policy.

5. Click **OK**.

6.1.4.9. Configure storage quota

If the capacity of a bucket reaches the specified storage quota, write operations such as PutObject, MultipartUpload, CopyObject, PostObject, and AppendObject cannot be performed on the bucket. This topic describes how to configure the storage quota of a created bucket.

Prerequisites

A bucket is created. For more information, see Create buckets.

Context

Take note of the following items when you configure the storage quota of a bucket:

- Before you configure the storage quota of a bucket, make sure that the quota does not limit your business because write operations cannot be performed if the bucket capacity reaches the quota.
- In general, it takes about an hour for OSS to determine whether the bucket capacity exceeds the storage quota. In some cases, it can take longer.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure storage quota.
3. Click the **Basic Settings** tab, find the **Storage Quota** section.
4. Click **Configure**. Turn on **Storage Quota** and set **Storage Quota**.
   - Units: TB or GB.
   - Valid values: -1 to 2000000

     The default value is -1, which indicates that the bucket capacity is not limited.
5. Click **Save**.

6.1.4.10. Configure back-to-origin rules

If you access data in a bucket that has no back-to-origin rules configured and the data does not exist, 404 Not Found is returned. However, if you configure back-to-origin rules that contain the correct origin URL, you can obtain the data based on the back-to-origin rules.

Context

Back-to-origin supports the mirroring and redirection modes. You can configure back-to-origin rules for hot migration and specific request redirection.

- Mirroring-based back-to-origin

  After mirroring-based back-to-origin rules are configured for a bucket, you can obtain an object in the bucket based on the rules when the requested object is not found. For example, when you perform the GetObject operation on an object and the object is not found, OSS retrieves the object based on the origin URL, returns the object, and then writes the object to OSS. Mirroring-based back-to-origin rules are used to seamlessly migrate data to OSS. This feature allows you to migrate a service that already runs on a user-created origin or in another cloud service to OSS without interrupting services.

- Redirection-based back-to-origin

  After redirection-based back-to-origin rules are configured for a bucket, you can obtain an object in the bucket based on the rules when the requested object is not found. For example, when you perform the GetObject operation on an object and the object is not found, OSS redirects the request to the origin URL, and then a browser or client returns the content from the origin. You can use this feature to redirect requests for objects and develop various services based on redirection.

You can configure up to 20 back-to-origin rules, which are run in a sequence that they are configured.

> **Notice** Back-to-origin rules and versioning cannot be configured for a bucket at the same time.

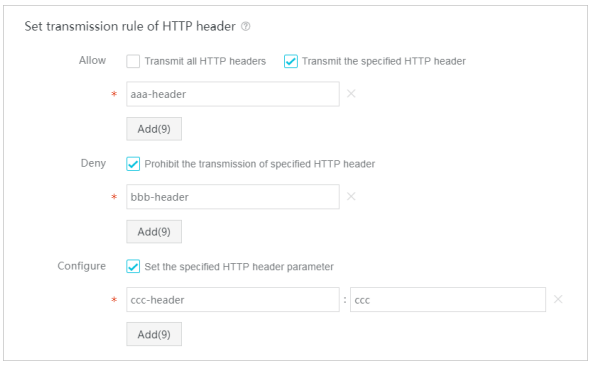
Configure a mirroring-based back-to-origin rule

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure the mirroring-based back-to-origin rule.
3. Click the **Basic Settings** tab. In the **Back-to-Origin** section, click **Configure**.
4. Click **Create Rule**. In the **Create Rule** panel, set the parameters described in the following table to create a mirroring-based back-to-origin rule.

Parameter	Required	Description
Mode	Yes	Select **Mirroring**. In this mode, when a requested object cannot be found in OSS, OSS automatically retrieves the object from the origin, stores the object in OSS, and then returns the content to the requester.
Prerequisite	Yes	Configure the conditions that trigger the back-to-origin rule. A rule is triggered only when all conditions are met. ○ **HTTP Status Code**: The back-to-origin rule is triggered when the specified HTTP status code is returned. The default HTTP status code is 404, which indicates that the rule is triggered when the requested object is not found in OSS and HTTP status code 404 is returned. By default, this option is selected when you select the **Mirroring** mode. ○ **File Name Prefix**: The back-to-origin rule is triggered when the name of the requested object contains the specified prefix. For example, when this parameter is set to *abc/*, the back-to-origin rule is triggered when you access `https://bucketname.endpoint/abc/image.jpg`. ○ **File Name Suffix**: The back-to-origin rule is triggered when the name of the requested object contains the specified suffix. For example, when this parameter is set to *.jpg*, the back-to-origin rule is triggered when you access `https://bucketname.endpoint/image.jpg`. **Note** File Name Prefix and File Name Suffix are optional when you configure only one back-to-origin rule. When you configure multiple back-to-origin rules, you must differentiate the rules by specifying a different prefix or suffix for each of the rules.
Replace or Delete File Prefix	No	You can configure this parameter after you select and configure **File Name Prefix**. When OSS sends a request to the origin, the content of **File Name Prefix** is replaced with that of **Replace or Delete File Prefix**. Example: You want to store the object obtained from the origin in the *mirror* folder under the root folder of the bucket. If the name of the requested object is *path/test/photo.jpg*, set **File Name Prefix** to *mirror/*, **Replace or Delete File Prefix** to *test/*, and the third column of **Origin URL** to *path*. In this case, when you access `https://bucketname.endpoint/mirror/photo.jpg`, if *photo.jpg* does not exist, OSS sends the `https://origin URL/path/test/photo.jpg` request to obtain this object. If the object is obtained from the origin, OSS stores this object in the *mirror/* folder and returns the object to you.

Parameter	Required	Description
Origin URL	Yes	Configure the information about the origin URL. ◦ First column: Select HTTP or HTTPS based on the protocol used by the origin. **Note** If SNI is enabled on the origin, HTTPS that you select does not take effect. ◦ Second column: Enter the domain name or IP address of the origin. Internal endpoints and IP addresses are not supported. If you enter an IP address, ensure that the origin corresponding to the IP address can be accessed by using the IP address. ◦ Third column: Enter the folder where the requested object is stored. Separate subfolders in a folder with forward slashes (/). Example: `abc/123`.
MD5 Verification	No	If this option is selected, the MD5 hash of the object obtained from the origin is checked. When the response contains the Content-MD5 header value, OSS checks whether the MD5 hash of the object matches the Content-MD5 header value. ◦ If the MD5 hash of the object matches the Content-MD5 header value, the client obtains the object, and OSS saves the object by using mirroring-based back-to-origin. ◦ If the MD5 hash of the object does not match the Content-MD5 header value, OSS calculates the Content-MD5 header value of the object based on the data integrity and does not save the object. However, the client can obtain the object because the object is returned to the client.
Keep Forward Slash in Origin URL	No	OSS does not support object names that start with a forward slash (/). Therefore, when the name of the requested object starts with a forward slash (/), you must select this option to obtain the requested object based on the back-to-origin rules. For example, the origin is `https://www.example.com`, the name of the requested object is */object.txt*, and the name of the bucket is examplebucket. If this option is selected, the default public endpoint to access the requested object is `https://examplebucket.endpoint//object.txt`, and the origin URL is `https://www.example.com//object.txt`. The object is obtained from OSS, returned to the client, and then saved in the bucket as *object.txt*.
Other Parameter	No	If this option is selected, the query string included in the request to OSS is transferred to the origin.
3xx Response	No	If this option is selected, OSS follows the origin to direct the request to obtain the resource and stores the resource in buckets. If this option is not selected, OSS passes the 3xx response without obtaining the resource.
Set Transmission Rule of HTTP Header	No	You can configure the transmission rule for HTTP headers to customize transmission actions such as passthrough, filtering, or modification. For more information, see the following examples of transmission rule configurations for HTTP headers.

The following figure provides a sample transmission rule for HTTP headers.



You send a request that contains the following HTTP headers to OSS.

```
GET /object
host : bucket.oss-cn-hangzhou.aliyuncs.com
aaa-header : aaa
bbb-header : bbb
ccc-header : 111
```

After the mirroring-based back-to-origin rule is triggered, OSS sends the following request to the origin:

```
GET /object
host : source.com
aaa-header : aaa
ccc-header : ccc
```

Transmission rules do not support the following HTTP headers:

- Headers that contain the following prefixes:
  - x-oss-
  - oss-
  - x-drs-
- All standard HTTP headers. Examples:
  - content-length
  - authorization2
  - authorization
  - range
  - date

Configure a redirection-based back-to-origin rule

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure a redirection-based back-to-origin rule.
3. Click the **Basic Settings** tab. In the **Back-to-Origin** section, click **Configure**.
4. Click **Create Rule**. In the **Create Rule** panel, configure the parameters described in the following table to create a redirection-based back-to-origin rule.

Parameter	Required	Description

Parameter	Required	Description
Mode	Yes	Select **Redirection**. In this mode, OSS redirects requests that meet the prerequisites to the origin URL over HTTP, and then a browser or client returns the content from the origin to the requester.
Prerequisite	Yes	Configure the conditions that trigger the back-to-origin rule. A rule is triggered only when all conditions are met. ◦ **HTTP Status Code**: The back-to-origin rule is triggered when the specified HTTP status code is returned. The default HTTP status code is 404, which indicates that the rule is triggered when the requested object is not found in OSS and HTTP status code 404 is returned. ◦ **File Name Prefix**: The back-to-origin rule is triggered when the name of the requested object contains the specified prefix. For example, when this parameter is set to *abc/*, the back-to-origin rule is triggered when you access `https://bucketname.endpoint/abc/image.jpg`. ◦ **File Name Suffix**: The back-to-origin rule is triggered when the name of the requested object contains the specified suffix. For example, when this parameter is set to *.jpg*, the back-to-origin rule is triggered when you access `https://bucketname.endpoint/image.jpg`. **Note** File Name Prefix and File Name Suffix are optional when you configure only one back-to-origin rule. When you configure multiple back-to-origin rules, you must differentiate the rules by specifying a different prefix or suffix for each of the rules.
Replace or Delete File Prefix	No	You can configure this parameter after you select and configure **File Name Prefix**. When OSS sends a request to the origin, the content of **File Name Prefix** is replaced with that of **Replace File Name Prefix**. **Note** If you select this option, you can only set **Replace File Name Prefix** for **Origin URL**.

Parameter	Required	Description
Origin URL	Yes	Configure the information about the origin URL. You must configure the back-to-origin rule based on the origin URL. Configuring the origin URL in accordance with the following rules: ○ First column: Select HTTP or HTTPS based on the protocol used by the origin. ○ Second column: Enter the domain name or IP address of the origin. **Note** You can enter IP addresses only when the origin can be directly accessed by using IP addresses. ○ Third column: Configure the redirection-based back-to-origin rule based on the prefix and suffix configurations. ■ **Add Prefix or Suffix**: Add a prefix or suffix to the redirected URL. The prefix is configured in the third column. The suffix is configured in the fourth column. You can select this option to configure information for the redirected URL when the URL of the requested data excludes a prefix or suffix. For example, the prefix is set to *123/*, and the suffix is set to *.jpg*. When you access `https://bucketname.endpoint/image`, the request is redirected to `https://Origin URL/123/image.jpg`. ■ **Redirect to Fixed URL**: Access to the requested object is redirected to a specified object. The object address is specified in the third column. Example: `abc/myphoto.jpg`. **Note** You can also set the value of this parameter to the URL of a website. For example, if you set the URL to `https://www.aliyun.com/index.html`, access to the bucket is redirected to the Alibaba Cloud homepage. ■ **Replace File Name Prefix**: If you set this parameter and **File Name Prefix** for Prerequisite, the prefix in the name of the object to which the access is redirected is replaced with that of the third column. If you set this parameter without setting **File Name Prefix** for Prerequisite, the specified prefix is added to the name of the object to which the access is redirected. Separate subfolders in a folder with forward slashes (/). The folder name must end with a forward slash (/). The folder name cannot contain asterisks (*).
Other Parameter	No	If this option is selected, the query string included in the request to OSS is transferred to the origin.
Redirection Code	Yes	You can select the redirection code from the drop-down list. Select **Source from Alibaba Cloud CDN** if the redirect request is from Alibaba Cloud CDN.

5. Click **OK**.

6.1.4.11. Configure server-side encryption

OSS supports server-side encryption. When you upload an object to a bucket for which server-side encryption is enabled, OSS encrypts the object and stores the encrypted object. When you download the encrypted object from OSS, OSS automatically decrypts the object and returns the decrypted object to you. A header is added in the response to indicate that the object is encrypted on the OSS server.

Context

OSS supports the following encryption methods:

- Server-side encryption by using KMS (SSE-KMS)

  OSS uses the default customer master key (CMK) managed by KMS or a specified CMK to encrypt objects. The CMK is managed by KMS to ensure confidentiality, integrity, and availability at minimal costs.

- Server-side encryption by using OSS-managed keys (SSE-OSS)

  OSS uses data keys to encrypt objects and manages the data keys. In addition, OSS uses master keys that are regularly rotated to encrypt data keys.

You can enable server-side encryption in the OSS console by using one of the following methods:

- Method 1: Enable server-side encryption when you create a bucket
- Method 2: Enable server-side encryption on the Basic Settings tab

Method 1: Enable server-side encryption when you create a bucket

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click **Create Bucket**.
3. On the **Create OSS Bucket** page, set parameters.

   You can set the following parameters to configure server-side encryption for the bucket.

   - **Server-Side Encryption**: Specify the encryption methods.
     - **None**: Server-side encryption is not performed.
     - **AES256**: AES256 is used to encrypt each object in the bucket by using different data keys. The CMKs used to encrypt the data keys are rotated regularly.
     - **SM4**: SM4 is used to encrypt each object in the bucket by using different data keys. The CMKs used to encrypt the data keys are rotated regularly.
     - **KMS**: CMKs managed by KMS are used to encrypt objects in the bucket.
   - **Encryption Algorithm**: This parameter can be configured when you select **KMS** for **Server-Side Encryption**. You can select **SM4** or **AES256**.
   - **Key ID**: This parameter can be configured when you select **KMS** for **Server-Side Encryption**. OSS uses the specified CMK to encrypt objects in the bucket.

     > **Note** To select a CMK ID for server-side encryption, you must create the CMK in the KMS console.

4. Click **Submit**.

Method 2: Enable server-side encryption on the Basic Settings tab

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure server-side encryption.
3. Click the **Basic Settings** tab. Find the **Server-side Encryption** section.
4. Click **Configure** and set the following parameters:

- **Encryption Method**: Specify the encryption method.
  - **None**: Server-side encryption is not performed.
  - **OSS-Managed**: Keys managed by OSS are used to encrypt your data.
  - **KMS**: CMKs managed by KMS are used to encrypt objects in the bucket.
- **Encryption Algorithm**: You can select **SM4** or **AES256**.
  - **AES256**: AES256 is used to encrypt each object in the bucket by using different data keys. CMKs used to encrypt the data keys are rotated regularly.
  - **SM4**: SM4 is used to encrypt each object in the bucket by using different data keys. CMKs used to encrypt the data keys are rotated regularly.
- **CMK**: This parameter can be configured when you select **KMS** for **Encryption Method**. OSS uses the specified CMK to encrypt objects in the bucket.

  > **Note** To select a CMK ID for server-side encryption, you must create the CMK in the KMS console.

5. Click **Save**.

> **Notice** The configurations of the default encryption method for a bucket do not affect the encryption configurations of existing objects within the bucket.

6.1.4.12. Bind a bucket to a VPC network

You can bind your bucket to a specified virtual private cloud (VPC) network to allow only requests from IP addresses within the VPC network to access your bucket.

Prerequisites

A VPC network is created. For more information, see the "Create a VPC" chapter of *VPC User Guide*.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket that you want to bind to the VPC network.
3. Click the **Overview** tab. Click **Bind VPC** in the **VPC Info** section.
4. On the **Bind VPC** page, select the VPC network that you create.

   You can also click **Create VPC** to create a new VPC network.
5. Click **Submit**.

6.1.4.13. Configure CRR

Cross-region replication (CRR) enables the automatic and asynchronous (near real-time) replication of objects across buckets in different OSS regions. Operations such as the creation, overwriting, and deletion of objects can be synchronized from the source bucket to the destination bucket.

Prerequisites

The steps described in Create buckets are performed or a bucket is created in the region.

Context

This feature meets the requirements of geo-disaster recovery or data replication. Objects in the destination bucket are extra replicas of objects in the source bucket. They have the same object names, object content, and

object metadata such as the creation time, owner, user metadata, and object ACL.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click the name of a bucket for which you want to configure CRR to go to the bucket details page.
3. On the bucket details page, click the **Basic Settings** tab. Find the **Cross-Region Replication** section.
4. Click **Enable**. In the **Cross-Region Replication** dialog box that appears, configure the parameters described in the following table.

Parameter	Description
Source Region	The region where the current bucket is located.
Source Bucket	The name of the current bucket.
Destination Region	Select the region where the destination bucket is located. The source and destination buckets for CRR must be located in different regions. Data cannot be synchronized between buckets located within the same region.
Destination Bucket	Select the destination bucket. The two buckets with CRR enabled cannot synchronize data with other buckets. If you synchronize data from Bucket A to Bucket B, neither Bucket A nor Bucket B can synchronize data with other buckets.
Applied To	Select the source data to synchronize. ○ **All Files in Source Bucket**: synchronizes all objects from the source bucket to the destination bucket. ○ **Files with Specified Prefix**: synchronizes the objects whose names contain the specified prefix from the source bucket to the destination bucket. For example, if you have a folder named *management/* in the root folder of a bucket and a subfolder named *abc/* in *management/*, when you want to synchronize objects in the *abc/* subfolder, enter *management/abc/* as the prefix. You can specify up to five prefixes.
Operations	Select the synchronization policy. ○ **Add/Change**: synchronizes only added or changed data from the source bucket to the destination bucket. ○ **Add/Delete/Change**: synchronizes all data changes such as the creation, overwriting, and deletion of objects from the source bucket to the destination bucket.
Replicate Historical Data	Specify whether to synchronize historical data before you enable CRR. ○ **Yes**: synchronizes historical data to the destination bucket. **Notice** When historical data is synchronized, objects in the source bucket may overwrite objects in the destination bucket if these objects have the same name. Before you select this option, ensure that the data is consistent. ○ **No**: synchronizes only objects that you want to upload or update after CRR is enabled to the destination bucket.

5. Click **OK**.

> Note
>
> - After the configuration is complete, it may take three to five minutes for CRR to take effect. Synchronization information is displayed after the source bucket is synchronized.
> - In CRR, data is asynchronously (near real-time) replicated. It takes several minutes to several hours for the data to be replicated to the destination bucket based on the amount of data.

6.1.5. Objects

6.1.5.1. Search for objects

You can search for objects whose names contain specific prefixes in buckets or folders in the OSS console.

Prerequisites

Object are uploaded to the bucket. For more information, see Upload objects.

Context

When you search for objects based on a prefix, search strings are case-sensitive and cannot contain forward slashes (/). You can search for objects only in the root folder of the current bucket or in the current folder. Subfolders and objects stored in subfolders cannot be searched.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket in which the objects that you want to search for are stored.
3. Click the **Files** tab.
4. Search for objects.
   - Search for objects or folders within the root folder of the bucket

     In the upper-right corner, enter the prefix to search in the search box and press Enter or click the icon to search for related objects. Objects and subfolders whose names contain the specified prefix within the root folder of the bucket are displayed.
   - Search for objects or subfolders within a specified folder

     Click the folder in which the objects or subfolders that you want to search for are stored. In the upper-right corner, enter the prefix to search in the search box and press Enter or click the icon to search for related objects. Objects and subfolders whose names contain the specified prefix within the current folder are displayed.

6.1.5.2. Configure object ACLs

You can configure the ACL of an object in the OSS console to control access to the object.

Prerequisites

An object is uploaded to the bucket. For more information, see Upload objects.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket that

contains the object whose ACL you want to configure.

3. In the left-side navigation pane, click **Files**.
4. Click the name of the object whose ACL you want to configure. In the **View Details** panel, click **Set ACL** on the right side of **File ACL**.

   You can also choose **More > Set ACL** in the Actions column corresponding to the object whose ACL you want to configure.
5. In the **Set ACL** panel, configure the ACL of the object.

   You can set the ACL of the object to one of the following values:

   - **Inherited from Bucket**: The ACL of the object is the same as that of the bucket.
   - **Private**: Only the owner or authorized users of this bucket can read and write objects in the bucket. Other users, including anonymous users cannot access the objects in the bucket without authorization.
   - **Public Read**: Only the owner or authorized users of this bucket can write objects in the bucket. Other users, including anonymous users can only read objects in the bucket.
   - **Public Read/Write**: All users, including anonymous users can read and write objects in the bucket. Fees incurred by such operations are charged to the owner of the bucket. Exercise caution when you set the object ACL to this value.
6. Click **OK**.

6.1.5.3. Create folders

You can use the OSS console to create and simulate basic features of folders in Windows. This topic describes how to create a folder by using the OSS console.

Prerequisites

A bucket is created. For more information, see Create buckets.

Context

OSS does not use a hierarchical structure for objects, but instead uses a flat structure. All elements are stored in buckets as objects. To facilitate object grouping and to simplify management, the OSS console displays objects whose names end with a forward slash (/) as folders. These objects can be uploaded and downloaded. You can use OSS folders in the OSS console in the same manner as you use folders in Windows.

> **Note** The OSS console displays objects whose names end with a forward slash (/) as folders, regardless of whether these objects contain data. The objects can only be downloaded by calling an API operation or by using OSS SDKs.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket in which you want to create folders.
3. Click the **Files** tab. On the page that appears, click **Create Folder**.
4. In the **Create Folder** panel, enter the folder name.

   The folder name must comply with the following conventions:

   - The name can contain only UTF-8 characters and cannot contain emojis.
   - The name cannot start with a forward slash (/) or backslash (\). The name cannot contain consecutive forward slashes (/). You can use forward slashes (/) in a folder name to quickly create a subfolder. For example, when you create a folder named *example/test/*, the folder named *example/* is created in the root folder of the bucket and the subfolder named *test/* is created in the example/ folder.

- The name cannot be two consecutive periods (`..`).
- The folder name must be 1 to 254 characters in length.

5. Click **OK**.

6.1.5.4. Delete objects

You can delete uploaded objects in the OSS console when they are no longer needed.

Context

You can delete a single object or batch delete multiple objects. You can batch delete up to 100 objects. To delete specific objects or batch delete more than 100 objects, we recommend that you use API operations or OSS SDKs.

> **Notice** Deleted objects cannot be recovered. Exercise caution when you delete objects.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket in which the objects you want to delete are stored.
3. In the left-side navigation pane, click **Files**.
4. Select one or more objects that you want to delete in the object list, and then choose **Batch Operation > Delete**.

   You can also choose **More > Completely Delete** in the Actions column corresponding to the object you want to delete.
5. In the dialog box that appears, click **OK**.

6.1.5.5. Manage parts

When you use multipart upload to upload an object, the object is split into several parts. After all of the parts are uploaded to the OSS server, you can call CompleteMultipartUpload to combine the parts into a complete object.

Context

You can also configure lifecycle rules to clear parts that are not needed on a regular basis. For more information, see Manage lifecycle rules.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket in which the parts you want to delete are stored.
3. Click the **Files** tab. On the page that appears, click **Parts**.
4. In the **Parts** panel, delete the parts.
   - To delete all parts in the bucket, select all parts and then click **Delete All**.
   - To delete specific parts in the bucket, select these parts and then click **Delete**.
5. In the dialog box that appears, click **OK**.

6.1.5.6. Configure object tagging

You can configure object tagging to classify objects. Object tagging uses key-value pairs to identify objects. You can perform operations on multiple objects that have the same tag. For example, you can configure lifecycle rules for objects that have the same tag.

Context

Object tagging uses key-value pairs to identify objects. You can manage multiple objects that have the same tag. For example, you can configure lifecycle rules for objects that have the same tag or authorize RAM users to access objects that have the same tag.

Take note of the following items when you configure object tagging:

- You can add up to 10 tags to an object. The tags added to an object must have unique tag keys.
- A tag key can be up to 128 bytes in length. A tag value can be up to 256 bytes in length.
- Tag keys and values are case-sensitive.
- The key and value of a tag can contain letters, digits, spaces, and the following special characters:

  + - =. _ : /
- Only the bucket owner and authorized users have read and write permissions on object tags. These permissions are independent of object ACLs.
- In cross-region replication (CRR), object tags are also replicated to the destination bucket.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket in which the parts you want to delete are stored.
3. Click the **Files** tab.
4. Choose **More** > **Tagging** in the Actions column corresponding to the object to which you want to add tags.
5. In the **Tagging** panel, configure the **Key** and **Value** of the tag.

   You can click **Add** to add up to more 10 tags to the object.
6. Click **OK**.

6.1.6. Create single tunnels

You can create single tunnels between OSS and a virtual private cloud (VPC) to access OSS resources from the VPC.

Prerequisites

A VPC and a vSwitch are created.

For more information, see the *Create a VPC* and *Create a vSwitch* topics in *VPC User Guide*.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation panel, click **Create Single Tunnel**.
3. Click **Create**.
4. On the **Create Single Tunnel** page, configure the parameters described in the following table.

Parameter	Required	Description
Organization	Yes	Select the organization of the VPC from which you want to access OSS resources.

Parameter	Required	Description
Resource Set	Yes	After you select an organization, the resource set is automatically selected based on the organization.
Region	Yes	After you select an organization, a region is automatically selected based on the organization.
Description	No	Enter the description of the single tunnel you want to create. The description cannot exceed 180 characters in length.
VPC	Yes	Select the VPC that you created. You can also click **Create VPC** to create a VPC.
vSwitch	Yes	Select the vSwitch that you created. You can also click **Create vSwitch** to create a vSwitch.

5. Click **Submit**.

6.1.7. Add OSS paths

You can add the paths of OSS resources in the console for quicker access.

Prerequisites

A bucket is created. For more information about how to create a bucket, see Create buckets.

Procedure

1. Log on to the OSS console.
2. Click the + icon on the right side of **My OSS Paths**.
3. In the **Add Authorized OSS Path** panel, add a path.

   You can configure the following parameters to add a path.

   - **Region**: Select the region of the bucket in the path that you want to add.
   - **File Path**: Add the path of the resource that you want to access. The path is in the `bucket/object-prefix` format. For example, if the OSS resource that you want to access is the root folder of a bucket named *example*, set File Path to *example*. If the OSS resource that you want to access is the *test* folder in the root folder of the bucket named *example*, set File Path to *example/test/*.

7.Apsara File Storage NAS

7.1. User Guide

7.1.1. What is NAS?

Apsara File Storage NAS is a cloud service that provides a file storage solution for compute nodes. The compute nodes include Elastic Compute Service (ECS) instances, Elastic High-Performance Computing (E-HPC) instances, and Container Service for Kubernetes (ACK) clusters.

NAS provides standard file access protocols. You can use the distributed file storage solution that provides shared access, scalability, high reliability, and high performance without the need to modify the configurations of the applications. You can mount a NAS file system on multiple compute nodes at a time. This reduces a large number of costs in data transmission and synchronization.

You can perform the following operations:

- Create a NAS file system and mount target.
- Create a permission group for the file system and add rules to the permission group. This allows access from specific IP addresses or CIDR blocks to the file system. This also allows you to grant different levels of access permissions to the IP addresses or CIDR blocks.
- Mount the file system on compute nodes. The compute nodes include ECS instances and ACK clusters. NAS allows you to access the file system by using the Network File System (NFS) and Server Message Block (SMB) protocols. You can also call POSIX-based APIs to access the file system.
- Manage file systems, mount targets, and permission groups in the NAS console.
- Call NAS API operations to manage file systems.

7.1.2. Precautions

Before you use NAS, you must familiarize yourself with the following limits.

Limits on file systems

- Maximum number of files in a single file system: 1 billion.
- Maximum name length: 255 bytes.
- Maximum size of a single file: 32 TB.
- Maximum directory depth: 1,000 levels deep.
- Maximum capacity of a single file system: 10 PB for NAS Capacity and 1 PB for NAS Performance.
- Maximum number of compute nodes on which you can mount a single file system: 10,000. Note: The file system allows simultaneous access from the 10,000 compute nodes.
- Maximum size of a protocol packet: 4 MB.
- Maximum number of Change Notify requests: 512.

Limits on NFS clients

Limits on the usage of NFS clients are listed as follows.

- You can open a maximum of 32,768 files at a time on an NFS client. Files in the list folder and its subfolders are not counted as part of the total number of open files.
- Each unique mount on an NFS client can acquire a maximum of 8,192 locks across a maximum of 256 unique file or process pairs. For example, a single process can acquire one or more locks on 256 separate files, or 8 processes can each acquire one or more locks on 32 files.
- We recommend that you do not use an NFS client in a Windows environment to access an NFS file system.

Limits on SMB clients

Each file or folder can be opened a maximum of 8,192 times in parallel across compute nodes that each have a file system mounted and users that share access to each of these file systems. This represents a maximum of 8,192 active file handlers for each file system. A maximum of 65,536 active file handlers can exist on a file system.

Limits on the NFS protocol

- NAS supports the NFSv3 and NFSv4 protocols.
- NFSv4.0 does not support the following attributes: FATTR4_MIMETYPE, FATTR4_QUOTA_AVAIL_HARD, FATTR4_QUOTA_AVAIL_SOFT, FATTR4_QUOTA_USED, FATTR4_TIME_BACKUP, and FATTR4_TIME_CREATE. If one of the preceding attributes is applied to a file system, an NFS4ERR_ATTRNOTSUPP error appears on a client that has the file system mounted.
- NFSv4.1 does not support the following attributes: FATTR4_DIR_NOTIF_DELAY, FATTR4_DIR_NOTIF_DELAY, FATTR4_DACL, FATTR4_SACL, FATTR4_CHANGE_POLICY, FATTR4_FS_STATUS, FATTR4_LAYOUT_HINT, FATTR4_LAYOUT_TYPES, FATTR4_LAYOUT_ALIGNMENT, FATTR4_FS_LOCATIONS_INFO, FATTR4_MDSTHRESHOLD, FATTR4_RETENTION_GET, FATTR4_RETENTION_SET, FATTR4_RETENTEVT_GET, FATTR4_RETENTEVT_SET, FATTR4_RETENTION_HOLD, FATTR4_MODE_SET_MASKED, FATTR4_FS_CHARSET_CAP. If one of the preceding attributes is applied to a file system, an NFS4ERR_ATTRNOTSUPP error appears on a client that has the file system mounted.
- NFSv4 does not support the following operations: OP_DELEGPURGE, OP_DELEGRETURN, and NFS4_OP_OPENATTR. If one of the preceding operations is applied to a file system, an NFS4ERR_ATTRNOTSUPP error appears on a client that has the file system mounted.
- NFSv4 does not support delegations.
- The following issues are related to user IDs (UIDs) and group IDs (GIDs):
  - On Linux, mappings between UIDs or GIDs and usernames or group names are defined in configurations files. For NFSv3 file systems, if the mapping between an ID and a name is defined in a configuration file, the name is displayed. If no mapping can be found for a UID or GID, the UID or GID is displayed.
  - For NFSv4 file systems, the usernames and group names are displayed as nobody for all files if the version of a Linux kernel is earlier than 3.0. If the kernel version is later than 3.0, the rule used by NFSv3 file systems applies to display files.

> **Notice** If a file or directory is stored on an NFSv4 file system and the Linux kernel version is earlier than 3.0, we recommend that you do not use the chown or chgrp command. If you use either one of the commands, the UID and GID of the file or directory will change to nobody.

Limits on the SMB protocol

- NAS supports protocols including SMB 2.1 or later and operating systems including Windows 7, and Windows Server 2008 R2 or later. However, NAS does not support Windows Vista, or Windows Server 2008 or earlier. Compared with SMB 2.1 or later, SMB 1.0 has lower performance and functionality. Furthermore, Windows products that support SMB 1.0 are no longer offered or supported.
- Extended file attributes and client-side caching based on leases.
- Input/output control (IOCTL) or file system control (FSCTL) operations, such as creating sparse files, compressing files, retrieving NIC status, and creating reparse points.
- Alternate data streams.
- Identity authentication provided by Active Directory (AD) or Lightweight Directory Access Protocol (LDAP).
- Several features provided by SMB 3.0 or later, such as SMB Direct, SMB Multichannel, SMB Directory Leasing, and persistent handles.
- Access control lists (ACLs) on files or directories.

7.1.3. Quick start

7.1.3.1. Log on to the Apsara File Storage NAS console

This topic describes how to log on to the NAS console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, select **Products**. In the **Storage** section, click **Apsara File Storage NAS**.

7.1.3.2. Create a file system

This topic describes how to create a file system in the Apsara File Storage NAS console.

Context

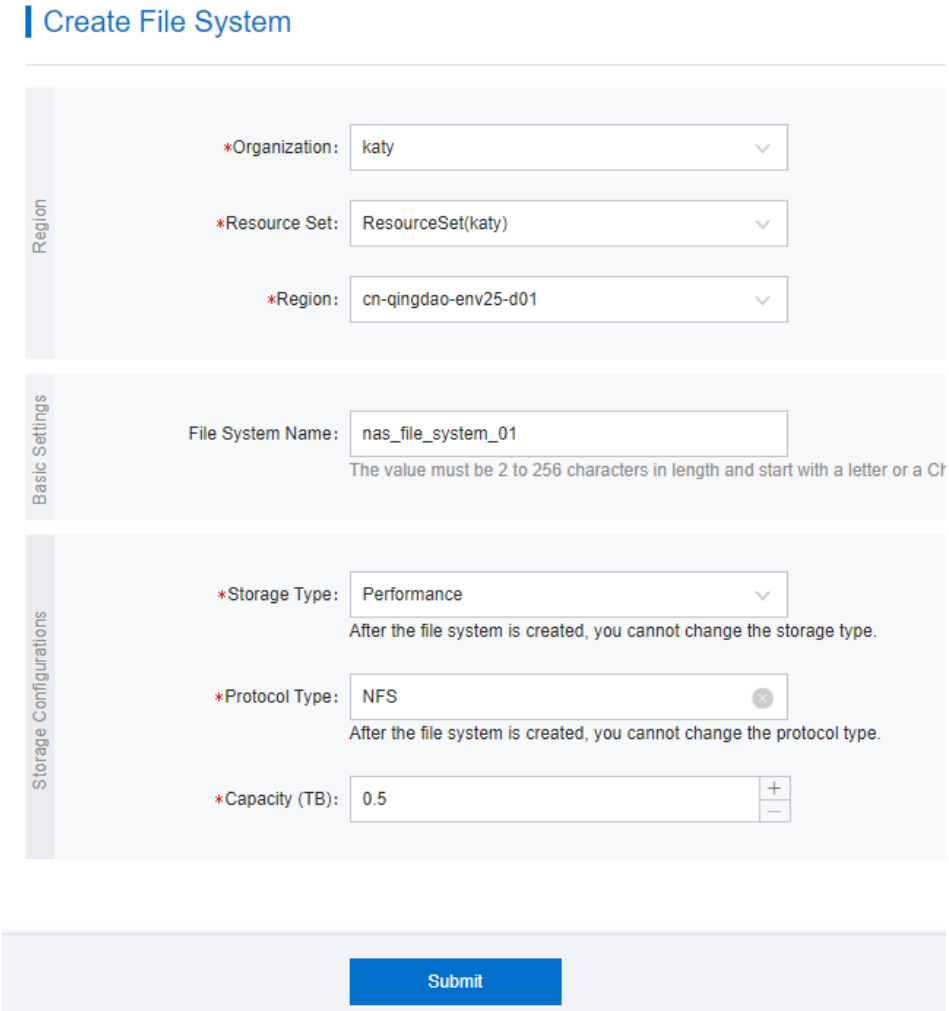
Before you create a file system, you must note the following limitations:

- You can use an Alibaba Cloud account to create a maximum of 1,000 file systems.
- The maximum capacity of a NAS Performance file system is 1 PB. The maximum capacity of a NAS Capacity file system is 10 PB.

  If you want to increase the maximum storage capacity, we recommend that you contact Alibaba Cloud Technical Support.

Procedure

1. Log on to the Apsara File Storage NAS console.
2. Choose **NAS > File System List** and click **Create File System**.
3. In the **Create File System** dialog box, set the required parameters.



The following table lists the required parameters.

Parameter	Description
Region	Select a region where you need to create a file system.
Organization	Select an organization from the drop-down list for the instance.
Resource Set	Select a resource set from the drop-down list for the instance.
File System Name	The name of the file system. The name must be 2 o 256 characters in length and can contain letters, digits, and special characters. These special characters include underscores (_) and hyphens (-). The name must start with a letter and cannot start with http:// or https://.
Storage Type	The storage type. Select **Performance** or **Capacity** based on your business requirements. The maximum capacity of an NAS Performance file system is 1 PB. The maximum capacity of an NAS Capacity file system is 10 PB.

Parameter	Description
Protocol Type	The protocol type. Select **NFS** or **SMB** based on your business requirements. We recommend that you mount Network File System (NFS) file systems on Linux clients and Server Message Block (SMB) file systems on Windows clients.
Capacity (TB)	The capacity of the file system. ○ The capacity of an NAS Performance file system ranges from 0.5 TB to 1024 TB. ○ The capacity of an NAS Capacity file system ranges from 0.5 TB to 10240 TB.

4. Click **OK** to complete the creation.

7.1.3.3. Create a permission group and add rules

This topic describes how to create a permission group and add rules to the permission group in the Apsara File Storage NAS console.

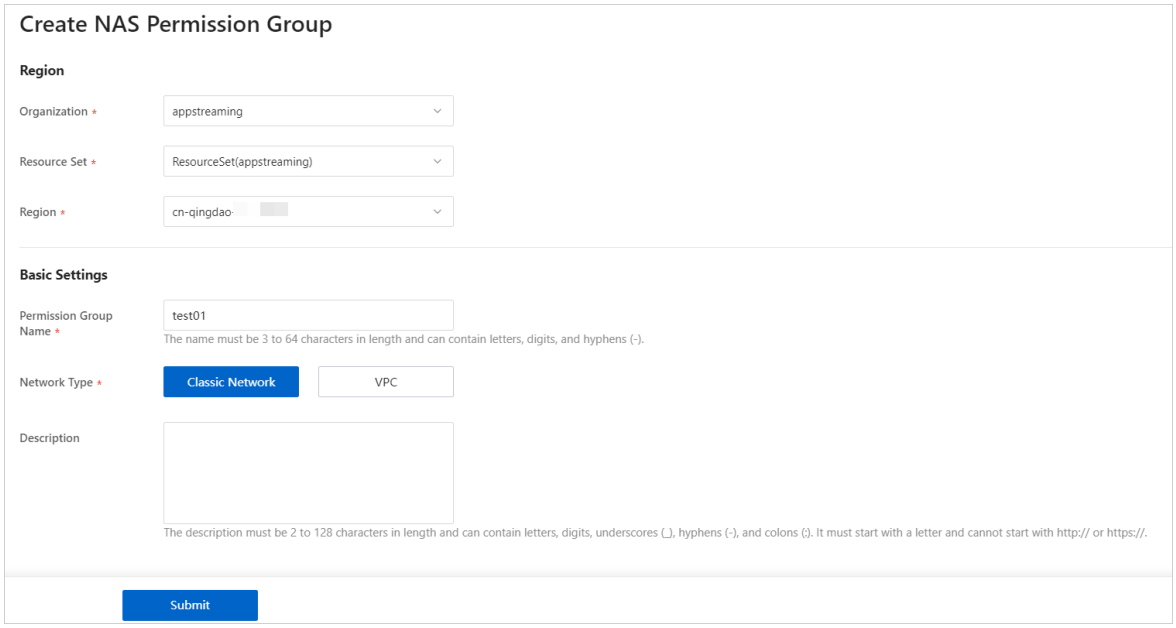
Context

In NAS, each permission group represents a whitelist. You can add rules to a permission group to allow access to a file system from specific IP addresses or CIDR blocks. You can also grant different access permissions to different IP addresses or CIDR blocks.

> **Note** You can use an Alibaba Cloud account to create a maximum of 100 permission groups. If you want to increase the limit, we recommend that you contact Alibaba Cloud Technical Support.

Creates a permission group

1. Log on to the Apsara File Storage NAS console.
2. Choose **NAS > Permission Group** and click **Create Permission Group**.
3. In the **Create Permission Group** dialog box, specify the required parameters.



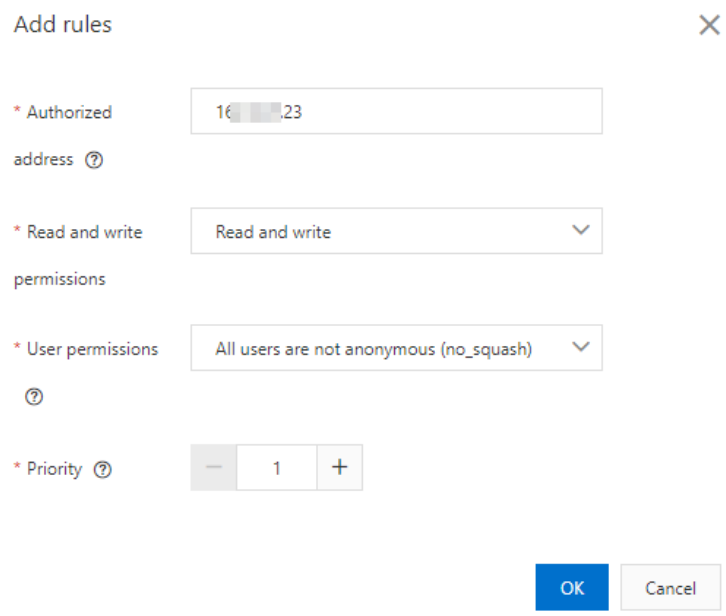
The following table lists the required parameters.

Parameter	Description
Organization	The organization to which the permission group belongs.
Resource Set	The resource set to which the permission group belongs.
Region	The region where you want to create the permission group.
Name	The name of the permission group. The name must be 3 to 64 characters in length and can contain letters, digits, and hypens (-).
Network Type	The network type. Select **Classic Network** or **VPC** based on your business requirements.

4. Click **OK** to complete the creation of the permission group.

Create a rule

1. Log on to the Apsara File Storage NAS console.
2. On the **Permission Group** page, find the target permission group and click **Manage**.
3. Click **Add Rule**.
4. In the **Add Rule** dialog box that appears, specify the required parameters.



The following table lists the required parameters.

Parameter	Description
Authorization Address	Specifies the authorized object to which the rule applies. You can specify an IP address or CIDR block. Only IP addresses are available for permission groups of the classic network type.
Read/Write Permission	Specifies whether to allow read-only or read/write access to the file system from the authorized object. Valid values: **Read-only** and **Read/Write**.

Parameter	Description
User Permission	Specifies whether to limit a Linux user's access to a file system. ○ **Do not limit root users (no_squash)**: allows access to a file system from root users. ○ **Limit root users (root_squash)**: denies access to a file system from root users. All root users are treated as nobody users. ○ **Limit all users (all_squash)**: denies access to a file system from all users including root users. All users are treated as nobody users. The nobody user is created by default on Linux. The user has only the most basic permissions and can access only the open content of servers. This feature offers high security.
Priority	When multiple rules are applied to an authorized object, the rule with the highest priority takes effect. Valid values: 1 to 100, in which 1 is the highest priority.

5. Click **OK** to complete the creation of the rule.

7.1.3.4. Add a mount target

This topic describes how to add a mount target. After an Apsara File Storage NAS file system is created, you must add a mount target to the file system. Then, you can use the mount target to mount the file system on compute nodes. These compute nodes include Elastic Compute Service (ECS) instances and Alibaba Cloud Container Service for Kubernetes (ACK) nodes.

Prerequisites

- A file system is created. For more information, see Create a file system.
- A permission group and a rule are created. For more information, see Create a permission group and add rules.

Context

A mount target is an endpoint that resides in a VPC or classic network. Each mount target corresponds to a file system. Mount targets of the VPC and classic network types are available for NAS file systems.

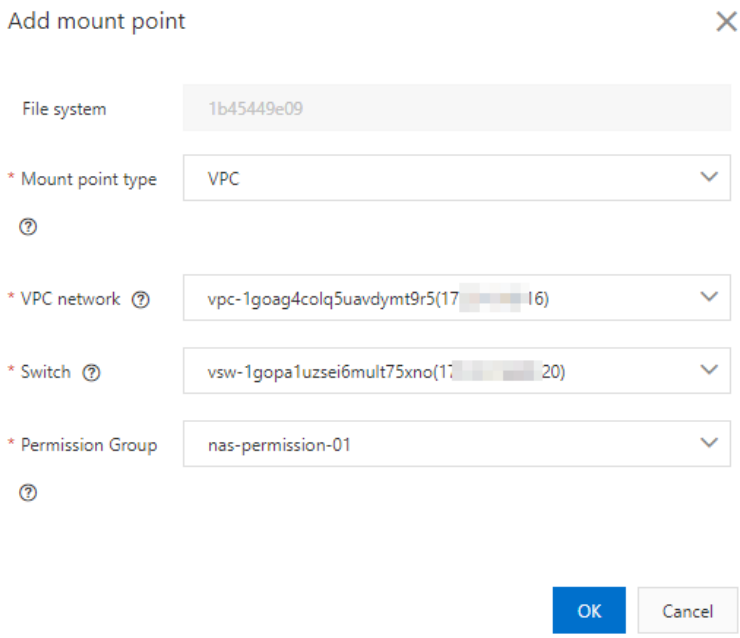
> **Note** You use a mount target to mount a file system on multiple compute nodes for shared access. These compute nodes include ECS instances and ACK nodes.

Procedure

1. Log on to the Apsara File Storage NAS console.
2. Choose **NAS > File System List**.
3. Find the target file system and click **Manage**.
4. On the **Mount Target** tab, click **Add Mount Target**.
5. In the **Add Mount Target** dialog box that appears, specify the required parameters.

   Mount Target Type: includes **VPC** and **Classic Network**.

   > **Note** Mount targets of the classic network type allow access only from ECS instances that belong to the same Alibaba Cloud account as the mount targets.



- If you want to create a mount target of the VPC type, specify the following parameters.

Parameter	Description
VPC	The VPC. **Note** The VPC you specify must be the same as the VPC where the compute nodes reside. These compute nodes include ECS instances and ACK nodes.
VSwitch	The VSwitch.
Permission Group	The permission group.

- If you want to create a mount target of the classic network type, specify the following parameters.

Parameter	Description
Permission Group	The permission group.

6. Click **OK** to complete the configuration.

7.1.3.5. Mount an NFS file system

This topic describes how to mount a Network File System (NFS) file system. Before you mount a file system, you must create the file system and a mount target for the file system. Then, you can use the mount target to mount the file system on compute nodes. These compute nodes include Elastic Compute Service (ECS) instances and Alibaba Cloud Container Service for Kubernetes (ACK) nodes.

Prerequisites

- A file system is created. For more information, see Create a file system.
- A permission group and a rule are created. For more information, see Create a permission group and add rules.
- A mount target is created. This topic takes a mount target of the VPC type as an example. For more

information, see Add a mount target.

- If you create a mount target of the VPC type for a file system, you can mount the file system only on ECS instances that reside in the same VPC as the mount target. You can specify a permission group for the mount target. Then, you can add several rules to the permission group. The authorization address of a rule must match the IP range of the VPC that hosts the ECS instances.
- If you create a mount target of the classic network type for a file system, you can mount the file system only on ECS instances that belong to the same Alibaba Cloud account as the mount target. You can specify a permission group for the mount target. Then, you can add several rules to the permission group. The authorization address of a rule must match the IP range of the private network that hosts the ECS instances.

- A compute node is created. This topic takes a Linux ECS instance as an example.

Step 1: Install an NFS client

Before you mount an NFS file system on a Linux ECS instance, you must install an NFS client. If an NFS client is instaned, skip this step.

1. Log on to the Linux ECS instance. For more information, see the **Quick start > Connect to an ECS instance** topic of the *ECS User Guide* .
2. Install the NFS client.
   - If CentOS, RHEL, or Aliyun Linux runs on the ECS instance, use the following command to install the NFS client.

     ```
     sudo yum install nfs-utils
     ```

   - If Ubuntu or Debian runs on the ECS instance, use the following commands to install the NFS client.

     ```
     sudo apt-get update
     ```

     ```
     sudo apt-get install nfs-common
     ```

Step 2: Mount an NFS file system

1. Log on to the Linux ECS instance. For more information, see the **Quick start > Connect to an ECS instance** topic of the *ECS User Guide* .
2. Mount the NFS file system.

   Use the following command to mount the NFS file system. In the command, replace file-system-id.region.nas.aliyuncs.com:/ /mnt with a value that is specific to your environment.

   - To mount an NFSv4 file system, use the following command.

     ```
     sudo mount -t nfs -o vers=4.0,minorversion=0,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport file-system-id.region.nas.aliyuncs.com:/ /mnt
     ```

   - To mount an NFSv3 file system, use the following command.

     ```
     sudo mount -t nfs -o vers=3,nolock,proto=tcp,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport file-system-id.region.nas.aliyuncs.com:/ /mnt
     ```

     Mount parameters

Parameter	Description

Parameter	Description
file-system-id.region.nas.aliyuncs.com:/ /mnt	Specifies the mount target of the NAS file system, the forward slash (/) following the mount target specifies the root directory of the NAS file system, and /mnt specifies a local directory that resides on the Linux ECS instance. You must replace the example values based on your business requirements. ■ The mount target, for example, file-system-id.region.nas.aliyuncs.com. To obtain information about a mount target, follow these steps. Log on to the NAS console, find the target system, click **Manage** next to the file system to go to the Details page. The Details page shows information about the mount target. ■ The directory of the NAS file system: specifies the root directory (/) or a subdirectory (/sub1). If a subdirectory is specified, make sure that the subdirectory exists. ■ The local directory on which you want to mount a file system: specifies the root directory (/) or a subdirectory (/mnt) of a system such as Linux. If a subdirectory is specified, make sure that the subdirectory exists.
vers	The version of the file system. Only NFSv3 and NFSv4 are available.
Mount option	When you mount a file system, multiple mount options are available. Separate multiple mount options with commas (,). For more information, see the following Mount options. **Note** When you specify mount options, take note of the following items. ■ To avoid a decrease in performance, we recommend that you specify the maximum value (1048576) for both the rsize mount option and the wsize mount option. ■ If you need to modify the timeo mount option, we recommend that you specify a minimum of 150 for the mount option. The timeo mount option is measured in deciseconds (tenths of a second). For example, a value of 150 indicates 15 seconds. ■ To avoid data inconsistency, we recommend that you do not use the soft mount option. Use caution with the soft mount option. ■ We recommend that you use the default values for other mount options. For example, a decrease in performance may occur due to changes in some mount options. These mount options include the size of the read or write buffer or the use of attribute caching.

Mount options

Option	Description
rsize	Specifies the maximum number of bytes in each read request that the NFS client can receive. Recommended value: 1048576.
wsize	Specifies the maximum number of bytes in each write request that the NFS client can send. Recommended value: 1048576.
hard	Specifies that applications must stop accessing a file system when the file system is unavailable, and wait until the file system is available. We recommended that you use the hard mount option.

Option	Description
timeo	Specifies the time in deciseconds (tenths of a seconds) that the NFS client waits before it retries an NFS request. Recommended value: 600.
retrans	Specifies the number of times the NFS client retries a request. Recommended value: 2.
noresvport	Specifies that the NFS client uses a different TCP source port for a new network connection to ensure data integrity. We recommend that you use the noresvport mount option.

3. Use the `mount -l` command to view the mount result.

   The following figure shows an example of a successful mount.

   After a file system is mounted, you can use the `df -h` command to view the size of the file system.

4. After you mount an NAS file system on an ECS instance, you can access the file system from the ECS instance.

   You can access the file system in the same way you access a local directory. The following figure shows an example.

7.1.3.6. Mount an SMB file system

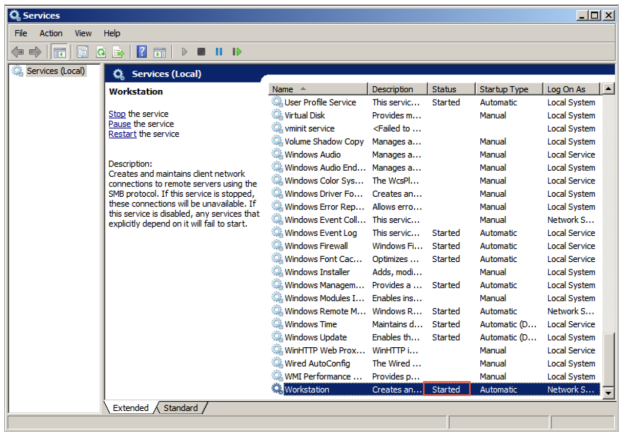
This topic describes how to mount a Server Message Block (SMB) file system. Before you mount a file system, you must create the file system and a mount target for the file system. Then, you can use the mount target to mount the file system on compute nodes such as Elastic Compute Service (ECS) instances.

Prerequisites

1. A file system is created. For more information, see Create a file system.
2. A permission group and a rule are created. For more information, see Create a permission group and add rules.
3. A mount target is created. This topic takes a mount target of the Classic Network type as an example. For more information, see Add a mount target.
   - If you create a mount target of the VPC type for a file system, you can mount the file system only on ECS instances that reside in the same VPC as the mount target. You can specify a permission group for the mount target. Then, you can add several rules to the permission group. The authorization address of a rule must match the IP range of the VPC that hosts the ECS instances.
   - If you create a mount target of the Classic Network type for a file system, you can mount the file system only on ECS instances that belong to the same Alibaba Cloud account as the mount target. You can specify a permission group for the mount target. Then, you can add several rules to the permission group. The authorization address of a rule must match the IP range of the private network that hosts the ECS instances.
4. An ECS instance is created. This topic takes a Windows ECS instance as an example.
5. The following Windows services are started:
   - Workstation
     a. Choose **All Programs > Accessories > Run**, or press `Win+R` and enter `services.msc` to open the Services console.

b. Find the Workstation service and ensure that the service is **Running** and the startup type is **Automatic**.

The default state for the Workstation service is Running.



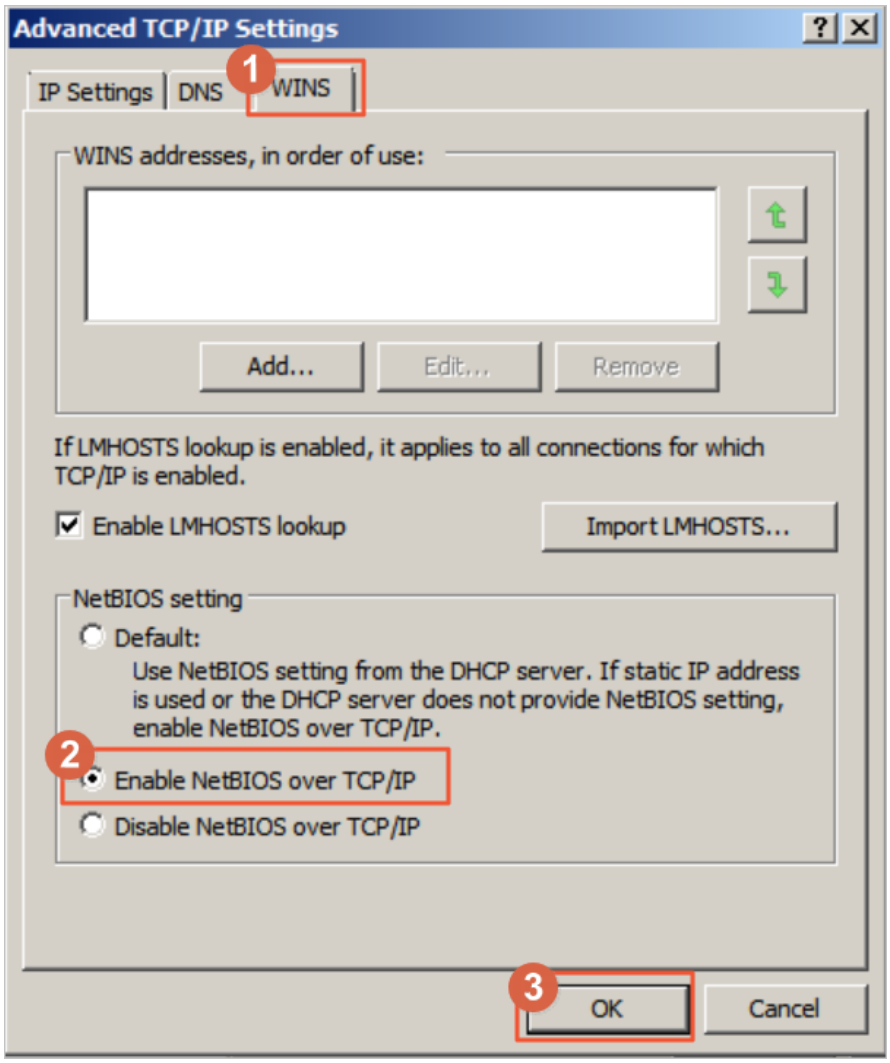
- TCP/IP NetBIOS Helper

Follow these steps to start the TCP/IP NetBIOS Helper service:

a. Double-click **Network and Sharing Center** and right-click the active network connection.

b. Click **Properties** to open the Local Area Network Properties dialog box. Double-click **Internet Protocol Version 4 (TCP/IPv4)** to open the Internet Protocol Version 4 (TCP/IPv4) Properties dialog box, and then click **Advanced**.

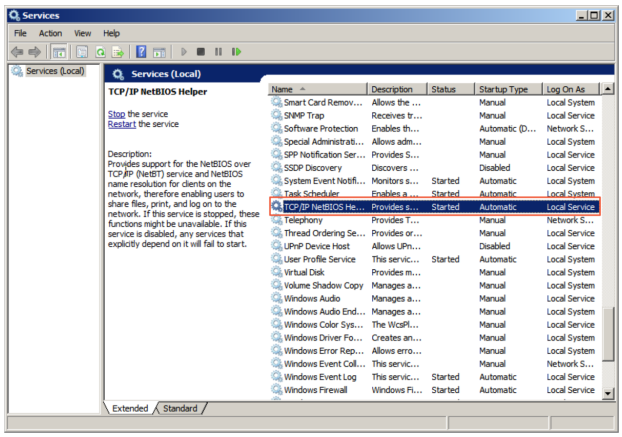
c. In the **Advanced TCP/IP Settings** dialog box, choose **WINS > Enable NetBIOS over TCP/IP**.



d. Choose **All Programs > Accessories > Run**, or press `Win+R` and enter `services.msc` to open the Services console.

e. Find the TCP/IP NetBIOS Helper service and make sure that the service is **Running** and the startup type is **Automatic**.

The default state for the TCP/IP NetBIOS Helper service is Running.



Procedure

1. Log on to a Windows ECS instance. For more information, see the **Quick start > Connect to an ECS instance** topic of the *ECS User Guide* .
2. Open the command prompt and use the following command to mount the file system.

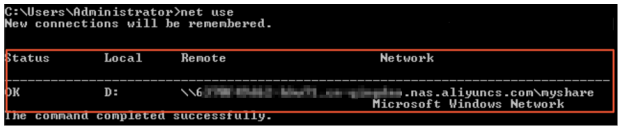
   ```
   net use D: \\file-system-id.region.nas.aliyuncs.com\myshare
   ```

   The syntax of the command is `net use <the letter of a local drive> \\<the domain name of a mount target>\myshare` .
   - The letter of a local drive: specifies a drive on which you need to mount a file system. You can specify the target drive based on your business requirements.

     > **Note** The target drive must be different from any existing drives.

   - The domain name of the mount target: When you create a mount target for a file system, the domain name of the mount target is automatically generated. You can replace the domain name based on your business requirements. To obtain the mount target of the file system, follow these steps. Log on to the NAS console, find the target file system, and click **Manage** to go to the Details page.
   - myshare: specifies the name of an SMB share. You cannot change the name.
3. Use the `net use` command to check mount results.

   The following figure shows an example of a successful mount.

4. After you mount an NAS file system on an ECS instance, you can access the file system from the ECS instance.

7.1.4. File systems

7.1.4.1. View the details of a file system

This topic describes how to view the details of a file system. The details include the information of the file system and the attached mount targets. You can view the details of a file system in the Apsara File Storage NAS console.

Prerequisites

A file system is created. For more information, see Create a file system.

Procedure

1. Log on to the Apsara File Storage NAS console.
2. Choose **NAS > File System List** .
3. Find the file system and click **Management** to go to the details page of the file system.

   The details page consists of the following sections:
   - **Basic Information**: In this section, the information of the file system is displayed. The information includes the ID, region, protocol type, and storage specifications.
   - **Mounting Use**: In this section, the list of mount targets attached to the file system is displayed. You can manage the mount targets and view the clients on which the file system is mounted.
   - **Quota Management** : In this section, the quota status of each directory in the file system is displayed. You can add, edit, and delete quotas for each directory.

7.1.4.2. Delete a file system

This topic describes how to delete a file system in the Apsara File Storage NAS console.

Prerequisites

A file system is created. For more information, see Create a file system.

Procedure

1. Log on to the Apsara File Storage NAS console.
2. Choose **NAS > File System List**.
3. Find the target file system, and click **Manage**.

   > **Note**
   > - Before you can delete a file system, you must remove all mount targets from the file system.
   > - Use caution when you delete a file system. After a file system is deleted, the data on the file system cannot be restored. We recommend that you ensure that all data is backed up.

4. In the **Delete File System** dialog box, click **OK** to complete the deletion.

7.1.4.3. Scale up a file system

If the used space of an Apsara File Storage NAS file system reaches the configured capacity, data can no longer be written to the file system. To ensure the availability of the NAS service, we recommend that you scale up the file system before the used space reaches the configured capacity. This topic describes how to scale up a file system in the NAS console.

Prerequisites

A file system is created. For more information, see Create a file system.

> **Notice** A file system can be scaled up but cannot be scaled down. A NAS Performance file system can be scaled up to a maximum of 1 PB. A NAS Capacity file system can be scaled up to a maximum of 10 PB.

Procedure

1.
2. In the left-side navigation pane, choose **File System > File System List**.
3. Find the file system and click **Upgrade** in the **Operations** column.
4. In the **Upgrade** dialog box, enter the scaled capacity of the file system in the **New Total Capacity** field.
5. Click **OK**.

7.1.5. Mount targets

7.1.5.1. View mount targets

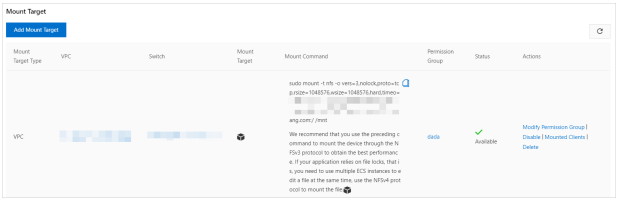
This topic describes how to view mount targets in the Apsara File Storage NAS console.

Context

- A file system is created. For more information, see Create a file system.
- A permission group and a rule are created. For more information, see Create a permission group and add rules.
- A mount target is created. For more information, see Add a mount target.

Procedure

1. Log on to the Apsara File Storage NAS console.
2. Choose **NAS > File System List**.
3. Find the target file system and click **Manage**.
4. In the **Mount Target** section, view mount targets in the file system.



7.1.5.2. Enable or disable a mount target

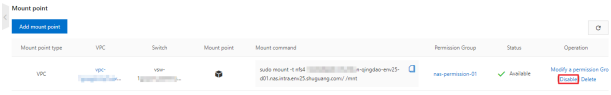
This topic describes how to enable or disable a mount target. You can control access to a mount target from clients by enabling or disabling the mount target.

Prerequisites

- A file system is created. For more information, see Create a file system.
- A permission group and a rule are created. For more information, see Create a permission group and add rules.
- A mount target is created. For more information, see Add a mount target.

Procedure

1. Log on to the Apsara File Storage NAS console.
2. Choose **NAS > File System List**.
3. Find the target file system and click **Manage**.
4. After you find the mount target that you want to disable or enable, you can perform the following operations:
   - Disable the mount target. Click **Disable**. In the Disable Mount Target dialog box, click **OK** to deny access to the mount target from clients.
   - Enable the mount target. Click **Enable**. In the Enable Mount Target dialog box, click **OK** to allow access to the mount target from clients.



7.1.5.3. Delete a mount target

This topic describes how to delete a mount target in the Apsara File Storage NAS console.

Prerequisites

- A file system is created. For more information, see Create a file system.
- A permission group and a rule are created. For more information, see Create a permission group and add rules.
- A mount target is created. For more information, see Add a mount target.

Procedure

1. Log on to the Apsara File Storage NAS console.

2. Choose **NAS > File System List**.
3. Find the target file system and click **Manage**.
4. Find the mount target that you want to delete and click **Delete**.

   > **Note** Use caution when you delete a mount target. After you delete a mount target, the mount target cannot be restored.

5. In the Delete Mount Target dialog box, click **OK**.

7.1.5.4. Modify the permission group of a mount target

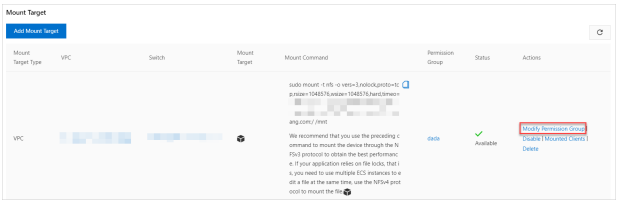
You must add a permission group to each mount target. You can modify the permission group of a mount target in the Apsara File Storage NAS console.

Prerequisites

- A file system is created. For more information, see Create a file system.
- A permission group and a rule are created. For more information, see Create a permission group and add rules.
- A mount target is created. For more information, see Add a mount target.

Procedure

1. Log on to the Apsara File Storage NAS console.
2. Choose **NAS > File System List**.
3. Find the target file system, and click **Manage**.
4. Find the mount target that you want to modify and click **Modify Permission Group**.



5. In the **Modify Permission Group** dialog box, change the permission group and click **OK**.

7.1.6. Permission groups

7.1.6.1. View permission groups

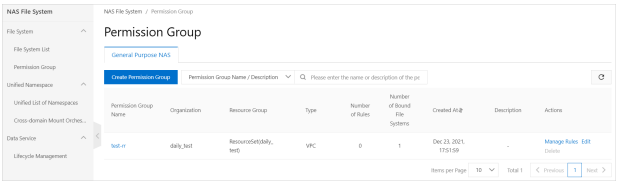
This topic describes how to view permission groups in the Apsara File Storage console.

Prerequisites

A permission group is created. For more information, see Create a permission group and add rules.

Procedure

1. Log on to the Apsara File Storage NAS console.
2. Choose **NAS > Permission Group** and view permission groups in the region.



7.1.6.2. Delete a permission group

This topic describes how to delete a permission group in the Apsara File Storage NAS console.

Prerequisites

A permission group is created. For more information, see Create a permission group and add rules.

Operating system

1. Log on to the Apsara File Storage NAS console.
2. Choose **NAS > Permission Group**.
3. Find the target permission group and click **Delete**.

   > **Note** Permission groups in use cannot be deleted. Before you can delete a permission group, you must remove the permission group from the linked mount target.

4. In the Delete Permission Group dialog box, click **OK**.

7.1.6.3. Manage permission group rules

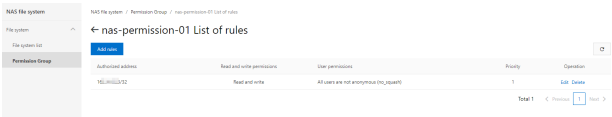
This topic describes how to manage permission group rules in the Apsara File Storage NAS console. The management includes viewing the details of rules, modifying rules, and deleting rules.

Prerequisites

A permission group and a permission group rule are created. For more information, see Create a permission group and add rules.

Procedure

1. Log on to the Apsara File Storage NAS console.
2. Choose **NAS > Permission Group**.
3. Find the target permission group and click **Manage**.
4. On the Rules page, you can perform the following operations:
   - View all rules for the permission group.

   - Modify a rule. Find the target rule and click **Edit** to modify the details of the rule. The details include the **authorization address**, **read/write permissions**, **user permission**, and **priority**.
   - Delete a rule. Find the target rule and click **Delete**. In the Delete Rule dialog box, click **OK**.

7.1.7. Manage quotas

This topic describes how to use the Alibaba Cloud quota_tool tool to manage quotas on Elastic Compute Service (ECS) instances that have Apsara File Storage NAS file systems mounted. You can configure, view, and cancel quotas on these ECS instances.

Prerequisites

An NFS file system of the NAS Capacity or NAS Performance type is mounted on an ECS instance. For more information, see Mount an NFS file system.

Context

NAS allows you to view and manage directory-level quotas. Directory-level quotas specify the maximum number of files in each directory and the maximum storage space for these files.

From the perspective of the application scope, quotas are sorted into quotas for all users and quotas for a single user or group. Quotas for all users specify the maximum storage space for files that all users can create in a directory. Quotas for a single user or group specify the maximum storage space for files that a user or group can create in a directory.

From the perspective of the restriction level, quotas are sorted into statistical quotas and restriction quotas. Statistical quotas collect only the usage of storage space. You can query and view statistical data. Restriction quotas specify the maximum capacity of storage space for files that you can create in a directory. If the limit is exceeded, you may fail to create a file or subdirectory, append data to a file, or perform other operations.

Notice

- Only statistical quotas are available.
- NAS performs asynchronous calculation for quotas at the backend. When you use the quota_tool tool to query statistical data about quotas, the process requires a period of time to complete. In most cases, the time period is about 5 to 15 minutes.

Configure quotas

This topic uses the */mnt* directory as an example.

1. Log on to an Elastic Compute Service (ECS) instance by using a root account.

   You can use the quota_tool tool on an ECS instance that has a NAS file system mounted. You must run the tool with the root permissions. The following describes how to use the quota_tool tool on the ECS instance.

2. Use the following command to download the quota_tool tool.

   ```
   wget https://nasimport.oss-cn-shanghai.aliyuncs.com/quota_tool_v1.0 -O quota_tool
   ```

3. Use the following command to grant the execute permission to the quota_tool tool.

   ```
   sudo chmod a+x quota_tool
   ```

4. Configure quotas.

   Note For each file system, you can configure quotas only for a maximum of 10 directories.

   The syntax of the command that you use to configure quotas is `sudo ./quota_tool set --dir [DIR] [OPTION]`.

Parameter	Description
--dir [DIR]	Specifies the directory for which you want to configure quotas. For example, --dir /mnt/data/.

Parameter	Description
OPTION	Specifies the required options. **Note** When you specify options, you must follow these rules: 1. the --accounting option is required. 2. One of the --alluser, --uid, and --gid options is required. . ○ --accounting: specifies a statistical quota. ○ --alluser: specifies a directory-level quota for all users. ○ --uid: specifies the UID of a user. For example, --uid 505 indicates the quota is configured only for the user whose UID is 505. ○ --gid: specifies the GID of a group. For example, --gid 1000 indicates the quota is configured only for the group whose GID is 1000.

The following examples describe how to configure quotas.

- Use the following command to configure a statistical quota for the */mnt/data/* directory to limit the total number of files that reside in a directory.

```
sudo ./quota_tool set --dir /mnt/data/ --accounting --alluser
```

- Use the following command to configure a statistical quota for the */mnt/data/* directory to limit the total number of files that can be created by the user whose UID is 505.

```
sudo ./quota_tool set --dir /mnt/data/ --accounting --uid 505
```

Query quotas

After you configure a quota for an NAS directory, you can query statistical data about the quota for the directory.

1. Log on to an Elastic Compute Service (ECS) instance by using a root account.
2. Use the following command to query quotas.

```
sudo ./quota_tool get --dir /mnt/data/ --all
```

In the preceding command, the --all parameter is optional. If you specify the parameter, statistical data about all quotas that are configured for the file system returns.

Note

- The first time you query a quota, a state called Initializing appears. After the **Initializing** process is complete, you can query the quota and a result showing **success** appears. The duration of the initialization process is based on the number of files and subdirectories in a directory.
- After the initialization process is complete, you can query quotas daily. A delay of 5 to 10 minutes may occur before the expected FileCountReal and SizeReal appear. This occurs due to the asynchronous calculation for quotas at the backend.

The following table lists parameters that are included in a response in the JSON format.

Parameter	Description
Path	Indicates a directory for which you query a quota.
Report	Includes all information about a quota that is specified for a directory, for example, UID and GID.
ReportStatus	The state for the query of a quota.
FileCountLimit	Indicates the limit for the number of files. A value of Empty indicates no limit.
FileCountReal	Indicates the total number of files including subdirectories, files, and special files that reside in a directory.
QuotaType	Accounting indicates a statistical quota and Force indicates a restriction quota.
Uid	Indicates the UID of a user. A value of All indicates all users.
Gid	Indicates the GID of a group. A value of All indicates all groups.
SizeLimit	Indicates the maximum capacity of files that reside in a directory. A value of Empty indicates no limit.
SizeReal	Indicates the total capacity of files that reside in a directory.

Cancel quotas

You can cancel a quota.

1. Log on to an ECS instance.
2. Use the following command to cancel quotas.

   The syntax of the command that you can use to cancel quotas is `sudo ./quota_tool cancel --dir [DIR] [OPTION]`.

Parameter	Description
--dir [DIR]	Specifies the directory for which you want to cancel quotas, for example, -dir /mnt/data/.

Parameter	Description
OPTION	Specifies the required options. **Note** When you specify the OPTION parameter, one of the --alluser, --uid, and --gid options is required. ○ --alluser: specifies a directory-level quota for all users. ○ --uid: specifies the UID of a user. For example, --uid 505 indicates that the quota is canceled for the user whose UID is 505. ○ --gid: specifies the GID of a group. For example, --gid 505 indicates that the quota is canceled for the group whose GID is 505.

The following examples describes how to cancel quotas.

- If you have configured a quota for the */mnt/data/* directory, use the following command to cancel the quota for the user whose UID is 100.

```
sudo ./quota_tool cancel --dir /mnt/data/ --uid 100
```

- If you have configured a quota for the */mnt/data/* directory, use the following command to cancel the quota for all users.

```
sudo ./quota_tool cancel --dir /mnt/data/ --alluser
```

7.1.8. Unified namespace

This topic describes how to create a unified namespace and create a mount target for the unified namespace in the Apsara File Storage NAS console. The topic also describes how to add, remove, and modify file systems in a namespace, view namespace details, and enable the cross-domain mount orchestration feature.

Features

A unified namespace allows you to mount multiple file systems in a NAS cluster by using a single domain name. You do not need to maintain multiple mount targets and mount directories.

You can create a mount target for a unified namespace. You can use a unified namespace to manage multiple file systems the same way you manage a single file system.

A unified namespace contains a virtual root directory in which file systems are the first-level subdirectories. After you add a file system to a unified namespace, you can still mount the file system by using the mount targets of the file system.

Limits

A unified namespace has the following limits:

- You can add a maximum of 1,000 file systems to each unified namespace.
- The mapping name of a file system in a unified namespace cannot exceed 255 characters in length. The name can contain only letters, digits, and the following special characters:

  .+-_()<>@#
- To enable a cross-domain mount orchestration, the mount directory name of a unified namespace cannot exceed 255 characters. The name can contain only letters, digits, and the following special characters:

  .+-_()<>@#
- A namespace supports the creation of up to two mount targets.

- You can create a maximum of 20 namespaces in each region.
- You can mount a unified namespace only by using the NFSv3 protocol.
- The file systems that are added to a namespace must belong to the same Alibaba Cloud account and cluster as the namespace. The storage type, protocol type, and encryption type of the file systems must be the same.
- The mapping name of a file system must be unique in each unified namespace.
- File systems can be mapped only to first-level subdirectories in a unified namespace. You cannot modify the access permissions, owner, or access control lists (ACLs).

Basic features of a unified namespace

- Create a unified namespace and create a mount target for the unified namespace.
  i. Log on to the Apsara File Storage NAS console.
  ii. In the left-side navigation pane, choose **Unified Namespace > Unified List of Namespaces**.
  iii. On the **Unified List of Namespaces** page, click **Create Namespace**. You can then create a namespace and a mount target for the namespace as prompted.

  > **Note** We recommend that you use different CIDR blocks if you create mount targets in virtual private clouds (VPCs) for unified namespaces in different regions. This identifies CIDR blocks when you mount unified namespaces across regions. For example, you can use `192.168.0.0/16` for Region 1 and `172.16.0.0/16` for Region 2. For more information, see Cross-domain mount orchestration.

- Add a file system to the unified namespace.

  After you create the unified namespace, you can add a file system and set the mapping name of the file system.

  > **Note** The mapping name is the name of the virtual directory for the file system.

- Remove the file system from the unified namespace.

  You can remove the existing file system from the unified namespace.

  > **Note** The file system is not deleted but removed from the file system list of the namespace.

- Modify the mapping name of the file system.

  You can modify the mapping name of the file system in the unified namespace.

  > **Note** If a symbolic or hard link exists between a file system and another file system, a connection failure may occur when you modify the mapping name of the file system.

- View the details of the unified namespace.

  The details of the namespace are divided into three sections:

  - Properties
  - Mount target list

    > **Note** You can create or delete mount targets.

  - File system list

    > **Note** You can add or remove file systems.

Cross-domain mount orchestration

In traditional solutions, you can add a file system to a namespace only if the file system and namespace reside in the same region. To add file systems to namespaces across regions, NAS provides the cross-domain mount orchestration feature. To use this feature, perform the following operations:

- Create a unified namespace and a mount target for the namespace.
- Map the unified namespace to the local directory tree of a client by using the cross-domain mount orchestration feature.
- After the orchestration is complete, specify the root directory to generate an automatic mount script.

> **Notice** The specified mount target and mapping path cannot be modified after the map is created. You can remove the map. However, if you create the map again, the specified mount target can only be attached to the original mapping path.

To mount the namespace on a client, you can run the automatic mount script on the client. This allows the client to access file systems across regions. The format of the local path on the client for each unified namespace is `<Specified root directory>/<Mapping path of the namespace>`.

To enable a cross-domain mount orchestration for different VPCs that reside in different regions, you must establish VPC connections. The following section describes how to establish VPC connections and enable a cross-domain mount by using an example.

1. Create a VPC.
   i. Log on to the Apsara File Storage NAS console.
   ii. In the left-side navigation pane, choose **Unified Namespace > Cross-domain Mount Orchestration**.
   iii. In the top navigation bar, choose **Products > Networking > Virtual Private Cloud**. On the VPCs page, click Create VPC.
   iv. On the **Create VPC** page, create two VPCs for Region 1 and Region 2.
      - Create a VPC named skvpc1 `192.168.1.0/24` for Region 1.
      - Create a VPC named skvpc2 `192.168.2.0/24` for Region 2.
2. Configure an Express Connect circuit across regions.

   Configure an Express Connect circuit between the VPCs of the two regions.

   Configure an Express Connect circuit in Region 1 to connect skvpc1 and skvpc2.
   i. Configure the route table for skvpc1.

      Add the CIDR block `192.168.2.0/24` of skvpc2 to the route table of skvpc1.
      a. In the left-side navigation pane, choose **VPCs > Route Tables**.
      b. On the **Route Tables** page, find the instance ID of skvpc1 and click **Manage** in the Actions column.
      c. On the Route Table page, click **Add Route Entry** and set the required parameters. Click **Create VPC-to-VPC Connection** to configure an Express Connect circuit between skvpc1 and skvpc2.

      > **Note**
      > - Specify the CIDR block of skvpc2 for **Destination CIDR Block**.
      > - Select **Router Interface (To VPC)** from the **Next Hop Type** drop-down list.
      > - If you configure an Express Connect circuit across regions for the first time, no VPCs are available.

ii. Create a VPC-to-VPC connection

Click Create VPC-to-VPC Connection to go to the **Create Peering Connection** page. Specify the source VPC ID, destination VPC ID, and bandwidth based on your business requirements.

After the VPC-to-VPC connection is created, you are redirected to the VPC-to-VPC page. If the initiator and acceptor are in the Activated state, the connection between skvpc1 and skvpc2 is established.

iii. View the VPC-to-VPC connection.

Return to the **Add Route Entry** page. Click the Refresh icon. The VPC-to-VPC connection that you have created appears in the VPC list.

Configure an Express Connect circuit in Region 2 to connect skvpc1 and skvpc2.

Perform the preceding procedure. In the Add Route Entry panel, select the VPC-to-VPC connection from the VPC list and click **OK**.

The route entry is then added to skvpc2 in Region 2. You can select the VPC-to-VPC connection from the VPC list because you have created a VPC-to-VPC connection. `vpc-6b xxxx42t` is the ID of skvpc1.

3. Add rules for NAS access groups.

   In the top navigation bar, choose **Products > Storage > Apsara File Storage NAS**. In the left-side navigation pane, choose **File System > Access Group**. On the **Access Group** page, click the permission group. On the page that appears, click **Add Rules**. In the dialog box that appears, set the authorized address and read and write permissions.

4. Create a mount target.

   In Region 2, you can create a mount target that resides in skvpc2 for the specified unified namespace.

   In Region 1, you can create a mount target that resides in skvpc1 for the specified unified namespace.

5. Create an Elastic Compute Service (ECS) instance.

   For example, if you select skvpc2 when you create an ECS instance, you can mount file systems on the ECS instance without the need to establish a VPC connection.

   > **Note** In most cases, each ECS instance uses an independent VPC. You do not need to establish connections between VPCs that are used by mount targets of namespaces in two regions. However, you must establish connections between the VPC that is used by mount targets of the namespace and the VPC that is used by the ECS instance.

7.1.9. Lifecycle management

This topic describes how to use the lifecycle management feature in the Apsara File Storage NAS console. The topic also describes how to configure lifecycle management policies and transfer cold data to the Infrequent Access (IA) storage medium based on the policies. In the NAS console, you can create, view, and modify lifecycle management policies. You can also query the usage of General-purpose NAS storage and IA storage mediums.

Prerequisites

A Network File System (NFS) file system of the NAS Capacity or NAS Performance type is mounted on an Elastic Compute Service (ECS) instance. For more information, see Mount an NFS file system.

Context

NAS provides the lifecycle management feature that allows you to manage cold data. You can configure lifecycle management policies and transfer infrequently accessed data to a lower-cost IA storage medium based on the policies. You can still access the data that is stored in the IA storage medium.

Limits

The lifecycle management feature supports only NFS file systems. Server Message Block (SMB) file systems and file systems whose data is encrypted are not supported.

Usage notes

The lifecycle management feature allows you to transfer cold data to a specified Object Storage Service (OSS) bucket. You must be the owner of the specified NAS file system and OSS bucket.

- You cannot delete the OSS bucket before you transfer cold data from the IA storage medium to the General-purpose storage. Otherwise, data loss may occur and the NAS cluster may become unavailable.
- You cannot revoke Resource Access Management (RAM) permissions from NAS.
- You must minimize the permissions of OSS bucket to prevent data leaks.
- We recommend that you use an independent OSS bucket for the lifecycle management feature of NAS. This prevents the risk of accidental deletion of cold data if you store the cold data with other business data in the same OSS bucket.

Preparations

1. Authorize NAS by using RAM.

   To use the lifecycle management feature, you must grant access permissions on OSS to NAS. NAS can then read data from and write data to specified OSS buckets.

   > **Notice** When you create a RAM role, you must specify an organization where the file system resides. In this example, the specified organization is bms.
   >
   > You need to grant permissions to NAS only once for an organization.

   i. Log on to the Apsara Uni-manager Management Console.

   ii. In the top navigation bar, choose **Configurations > RAM Service Linked Role**.

   iii. On the **RAM Roles** page, click **Create RAM Role** to create a RAM role and grant access permissions on OSS to the role.

   iv. View the authorization status of bms in the role list to confirm that AliyunNASTieringRole is in the list.

2. Create an OSS bucket.

   You can use an existing OSS bucket or create an OSS bucket. NAS allows you to configure multiple OSS buckets for each file system.

   > **Note** If you use an existing OSS bucket, the following requirements must be met:
   >
   > - The OSS bucket is in the first cluster of the region. The domain name of the first cluster starts with oss- whereas the domain names of the other clusters start with oss*xxxx*.
   > - The OSS bucket and the file system belong to the same organization.

   Log on to the Apsara Uni-manager Management Console. In the top navigation bar, choose **Products > Object Storage Service**. In the left-side navigation pane, choose **Buckets > Create Bucket**. On the Create OSS Bucket page, set the parameters and create an OSS bucket that is used to store cold data. In this example, an OSS bucket named nastiering is created.

   > **Note** When you create an OSS bucket, the following requirements must be met:
   >
   > - **Storage Class** is selected as **Standard** and the first cluster in the region is selected for the OSS cluster. The domain name of the first cluster starts with oss- whereas the domain names of the other clusters start with oss*xxxx*.
   > - The OSS bucket and the file system belong to the same organization.

Procedure

1. Create a lifecycle management policy.
   i. Log on to the Apsara Uni-manager Management Console.
   ii. In the top navigation bar, choose **Products > Apsara File Storage NAS**.
   iii. In the left-side navigation pane, click **Life Cycle Management**. On the page that appears, click Create Policy to create a lifecycle management policy.

   After you create a lifecycle management policy for a directory in the file system, NAS transfers cold data that meets the rule of the policy to the IA storage medium. When you create a lifecycle management policy, you can set the following parameters:

   - Policy Name: the name of the policy. The name must be unique within the policies.
   - File System: the file system for which the lifecycle management policy is configured.
   - Directory Path: the directory path on the file system. The path must start with a forward slash (/).

     You can enter a forward slash (/) to indicate the root directory.

     If you select **Recursive Subdirectory**, all subdirectories in the directory are recurred.
   - Management Rules: the pre-configured rules of the policy. You can select a rule to transfer files that have not been accessed more than 14, 30, 60, or 90 days to the IA storage medium.
   - OSS Bucket: the OSS bucket to which cold data is transferred.

     > **Note** Cold data in a file system can only be transferred to an OSS bucket.
     >
     > Only the OSS buckets of the organization where the file system resides are displayed in the OSS Bucket list.

2. View the lifecycle management policy.

   On the **Life Cycle Management** page, you can view the lifecycle management policy that you have created. You can also filter the policies by file system ID.

3. Modify the lifecycle management policy

   On the **Life Cycle Management** page, you can modify the lifecycle management policy.You can modify the following parameters:

   - Recursive Subdirectory
   - Management Rules

4. Query the usage of the General-purpose NAS storage and IA storage medium.

   In the left-side navigation pane, click **File System List**. You can query the usage of General-purpose NAS storage and IA storage medium within the file system for which you have configured the lifecycle management policy.

7.1.10. Directory-level ACLs that grant the read and write access

7.1.10.1. Overview

Apsara File Storage NAS supports NFSv4 access control lists (ACLs) and Portable Operating System Interface (POSIX) ACLs. This topic describes POSIX ACLs and NFSv4 ACLs. It also lists precautions for using these ACLs.

Access control and user management are important for enterprise-level users who want to share files between different users and groups by using a shared file system. To control access to different files and directories, you can grant users and groups different types of access. NAS provides Network File System (NFS) ACLs to allow you to meet specific requirements. An ACL consists of one or more access control entries (ACEs) that each grant a user or group one or more permissions to access a file or directory.

The NFSv3 protocol includes the extended support for POSIX ACLs. POSIX ACLs extend the support for access control over file mode creation masks. You can grant permissions to specific users and groups besides users of the owner, group, and other classes. Permissions can also be inherited from parent objects. For more information, see acl - Linux man page.

The NFSv4 protocol includes extended support for NFSv4 ACLs that provide more fine-grained access control than POSIX ACLs do. For more information, see nfs4_acl - Linux man page.

You can mount an NFSv3 file system that has NFSv4 ACLs applied. These NFSv4 ACLs will then be converted into POSIX ACLs. You can also mount an NFSv4 file system that has POSIX ACLs applied. These POSIX ACLs will then be converted into NFSv4 ACLs. If you use NFS ACLs, we recommend that you mount NFSv4 file systems and control access by using NFSv4 ACLs rather than file mode creation masks or POSIX ACLs. This is because: NFSv4 ACLs and POSIX ACLs are not fully compatible. The interaction between ACLs and file mode creation masks is not in an ideal state. The file systems that are mounted by using the NFSv3 do not support locks. For more information about NFS ACL features, see Features.

Precautions for using POSIX ACLs

- We recommend that you use the default inheritance method that allows a subdirectory or file to inherit the same ACL from the parent directory. This allows you to avoid configuring another ACL when you create a new file or subdirectory in the parent directory.
- We recommend that you retain a minimum number of ACEs because a file system needs to scan all ACEs each time it performs permission verification. Abuse of ACLs may diminish the performance of file systems.
- Use caution when you configure ACLs by using the recursive method (`setfacl -R`). Large amounts of metadata are produced when you perform a recursive operation on a directory that contains a large number of files and subdirectories. This may affect your businesses.
- Before you configure ACLs, we recommend that you manage groups and related permissions. For example, you can add a user to one or more groups. If you want to add, remove, or modify permissions for a user, we recommend that you move the user to a group that has the required permissions. You do not need to modify the ACL of a group as long as the structure of groups remains unchanged. We recommend that you configure ACLs for groups rather than single users. This provides a simple and effective time-saving method to control access and ensure the better organization of permissions.
- You can apply a POSIX ACL to multiple objects that resides on different clients. In such cases, you must ensure that the ACL you apply to each object is the same. Apsara File Storage NAS stores user IDs (UIDs) and group IDs (GIDs) at the backend. You must ensure that the mappings between a username or group name and a UID or GID are the same.
- We recommend that you grant the least permissions to the other class because all users have the permissions that are granted to the other class. A potential security vulnerability may be exposed if the other class has more permissions than any ACE.
- We recommended that you configure the least permissions for the other class. Before creating files or directories, you can use the `umask 777` command to configure the file mode creation mask. This command sets the file mode creation mask to 000 when the mask is used as a parameter to create a new file or directory. This ensures that the newly created file or directory has the least permissions. For more information, see umask and the default file mode creation mask.
- After you enable POSIX ACLs, the semantics of the other class for the POSIX ACL are equal to the semantics of the EVERYONE@ principal. The semantics of the other class for the file mode creation mask are also equal to the semantics of the EVERYONE@ principal. When a system performs permission verification, the system treats the other class the same as the EVERYONE@ principal.

Precautions for using NFSv4 ACLs

- Use UIDs or GIDs such as UID 1001 to configure ACLs.
- We recommend that you do not configure the file mode creation mask after you configure an NFSv4 ACL.
- The `nfs4_setfacl` command provides -a, -x, -m, and other options. You can use these options to add, remove, or modify ACEs. However, we recommend that you use `nfs4_setfacl -e <file>` the command to edit an ACL in an interactive mode.
- We recommend that you configure the least permissions for the EVERYONE@ principal because NFSv4 ACLs only support allow rather than deny ACEs. A potential security vulnerability may be exposed if the EVERYONE@ principal has more permissions than other ACEs.
- NFSv4 ACLs have fine-grained permissions. In most cases, it is unnecessary to subdivide permissions at such a fine-grained level. For example, if you have the write access (w) to a file but do not have the append-only (a) access, an error may occur when you write data to the file. The same issue occurs for a directory. To avoid unexpected permission errors, we recommend that you specify a capital w (W) as a parameter when you use the `nfs4_setfacl` command to configure an ACL. The `nfs4_setfacl` command converts W to a full write access permission. For a file, W is expanded to wadT. For a directory, W is expanded to wadTD.
- We recommend that you use the default inheritance method that allows a subdirectory or file to inherit the same ACL from the parent directory. This allows you to avoid configuring another ACL when you create a new file or subdirectory in the parent directory.
- We recommend that you retain a minimum number of ACEs because a file system needs to scan all ACEs each time it performs permission verification. Abuse of ACLs may diminish the performance of file systems.
- Use caution when you configure ACLs by using the recursive method (`nfs4_setfacl -R`). Large amounts of metadata are generated when you perform a recursive operation on a directory that contains a large number of files and subdirectories. This may affect your businesses.
- Before you configure ACLs, we recommend that you manage groups and related permissions. For example, you can add a user to one or more groups. If you want to add, remove, or modify permissions for a user, we recommend that you move the user to a group that has the required permissions. You do not need to modify the ACL of a group as long as the structure of groups remains unchanged. We recommend that you configure ACLs for groups rather than single users. This provides a simple and effective time-saving method to control access and ensure the better organization of permissions.

7.1.10.2. Features

This topic describes the features of NFSv4 access control lists (ACLs) and POSIX ACLs.

Features of Apsara File Storage NAS NFSv4 ACLs

- Only access control entries (ACEs) of the allow type are supported. ACEs of the following types are not supported: deny, audit, and alarm.

  Deny ACEs increase the complexity of access control. In most cases, complexity leads to confusion and increases potential security risks. As agreed by the industry, we recommend that you avoid using deny ACEs. For more information about why deny ACEs are not recommended, see FAQ.

  Audit and alarm ACEs are not available for NFS file systems. Instead, you can audit file systems and configure alerts based on auditing results in the NAS console.

- If no ACL is specified for a file or a directory, the default ACL that corresponds to the predefined file mode creation mask is applied.

```
touch file
```

```
[root@vbox test]# ls -l file
-rw-r--r--. 1 root root 0 May  6 14:27 file
```

```
[root@vbox test]# nfs4_getfacl file
# file: file
A::OWNER@:rwatTnNcCy
A::GROUP@:rtncy
A::EVERYONE@:rtncy
```

- An ACL is an ordered list that contains and deduplicates ACEs. This scheme ensures that permissions defined in an ACL are clear and informative.

  If you apply both a new ACE and an existing ACE to the same object and the existing ACE is inherited from the parent object, the permissions of the new ACE override the permissions of the existing ACE. For example:

  - In most cases, ACEs that include the following principals are queued in sequence at the beginning of an ACL: OWNER@, GROUP@, and EVERYONE@. These ACEs take precedence over other ACEs.

    ```
    [root@vbox test]# nfs4_getfacl file
    # file: file
    A::OWNER@:rwaxtTnNcCy
    A::GROUP@:rtncy
    A::EVERYONE@:rtncy
    A::1001:rwaxtTNcCy
    ```

  - Add an ACE of the read and write permissions to the following ACL for a user principal named 1009. The ACE is placed after the ACE that is defined for a user principal named 1001 based on the predefined order.

    ```
    [root@vbox test]# nfs4_setfacl -a A::1009:X file
    [root@vbox test]# nfs4_getfacl file
    # file: file
    A::OWNER@:rwaxtTcCy
    A::GROUP@:rwatcy
    A::EVERYONE@:tcy
    A::1001:rwaxtTNcCy
    A::1009:xtcy
    ```

  - Add a new ACE that includes the execute permission to the ACL for the user principal named 1009. The system automatically merges the execute permission into the existing ACE for the 1009 user principal.

    ```
    [root@vbox test]# nfs4_setfacl -a A::1009:W file
    [root@vbox test]# nfs4_getfacl file
    # file: file
    A::OWNER@:rwaxtTcCy
    A::GROUP@:rwatcy
    A::EVERYONE@:tcy
    A::1001:rwaxtTNcCy
    A::1009:waxtTncCy
    ```

- Add the f and d inheritance flags to an ACE that includes a user principal named 1009. Then, the system splits the ACE into two ACEs. One ACE has an extra inheritance flag named i specified, which indicates an inherit-only ACE. The other ACE only applies to the file object without inheritance flags. If the inheritance type of an existing ACE matches the type for one of the two ACEs, the system combines the existing ACE with the ACE out of the two ACEs. The two matching ACEs are combined into one ACE.

```
[root@vbox test]# nfs4_setfacl -a A:fd:1009:R file
[root@vbox test]# nfs4_getfacl file
# file: file
A::OWNER@:rwaxtTcCy
A::GROUP@:rwatcy
A::EVERYONE@:tcy
A::1001:rwaxtTNcCy
A::1009:rwaxtTNcCy
A:fdi:1009:r
```

- All ACEs can be inherited.
  i. For example, the OWNER@ principal has the write access, the GROUP@ principal has the read access, and the EVERYONE@ has no access to the dir directory.

```
[root@vbox nfs]# nfs4_getfacl dir
# file: dir
A::OWNER@:rwaDxtTnNcCy
A::GROUP@:rxtcy
A::EVERYONE@:tncy
```

  ii. Add an ACE that grants a user principal named 1000 the read, write, and execute access to the dir directory. The f and d inheritance flags are also specified for the ACE.

```
[root@vbox nfs]# nfs4_setfacl -a A:fd:1000:rwx dir
[root@vbox nfs]# nfs4_getfacl dir
# file: dir
A::OWNER@:rwaDxtTcCy
A::GROUP@:rxtcy
A::EVERYONE@:tcy
A::1000:rwx
A:fdi:1000:rwx
```

  iii. When you create a file or subdirectory in the dir directory, the file or the subdirectory automatically inherits all ACEs from the dir directory.

```
[root@vbox nfs]# touch dir/file
[root@vbox nfs]# nfs4_getfacl dir/file
# file: dir/file
A::OWNER@:rwatTcCy
A::GROUP@:rwatcy
A::EVERYONE@:rwatcy
A::1000:rwx
```

```
[root@vbox nfs]# mkdir dir/subdir
[root@vbox nfs]# nfs4_getfacl dir/subdir
# file: dir/subdir
A::OWNER@:rwaDxtTcCy
A::GROUP@:rwaDxtcy
A::EVERYONE@:rwaDxtcy
A:fdi:1000:rwx
```

> **Note**
> - We recommend that you grant the least privileges to the EVERYONE@ principal. Before you perform the following steps, we recommend that you run the `umask 777` command. This command ensures that no access to a file or directory is granted when the file or directory is created. For more information, see Why doesn't umask change execute permissions on files?
> - When Linux calls functions to create files or directory, the predefined file mode creation mask is used as a request parameter. You can obtain the final ACL for a child object from the overlap of the inherited ACL (parent to child) and the file mode creation mask, as specified in the RFC7530 standard. When you modify the group bits of a file mode creation mask based on the standard, permissions included in an ACL for each group must be less than or equal to permissions defined in group bits. However, this scheme results in an invalid inheritance for groups. For example, you create a file and the file attempts to inherit A:RWX from a parent object. However, the predefined file mode creation mask sets the group bits to R. The final permission for the file becomes A:R. In actual practice, we recommend that you only modify file mode creation masks for ACLs that include the following principals: OWNER@, GROUP@, and EVERYONE@. This prevents against potential issues and ensures that semantics are clear. To remove permissions for a group, we recommend that you remove the ACE that relates to the group.

- You need to manage mappings between usernames or group names and user IDs (UIDs) or group IDs (GIDs) across multiple independent instances.

  NAS NFS adopts IP security groups rather than usernames to authenticate users. When you configure NFSv4 ACLs, UIDs or GIDs that are included in ACEs are stored in Linux. When you print an ACL for an object in a shell, Linux automatically loads the */etc/passwd* file and converts UIDs or GIDs into usernames or group names. You need to manage mappings between usernames or group names and UIDs or GIDs across multiple instances. You must ensure a username or group name is mapped to its UID or GID.

- NFSv4 ACLs can be printed by using extended attributes.

```
[root@vbox nfs]# getfattr -n system.nfs4_acl file
# file: file
system.nfs4_acl=0sAAAABgAAAAAAAAAABYBhwAAAAZPV05FUkAAAAAAAAAAAAAAAABIAhwAAAAZHUk9VUEAAAAAAAAAAAAAAA
ABIAhwAAAAlFVkVSWU9ORUAAAAAAAAAAAAAAAAAAAAAAEAAAAEMTAwMAAAAAAAAAALAAAAAwAAAAQxMDAwAAAAAAAAAEAAFgGQAA
AABTEwMDAxAAAA
```

- Tools such as cp are supported for migrating NFSv4 ACLs.

  NAS allows you to migrate NFSv4 ACLs by using the cp, tar, and rsync tools. For more information, see How to preserve NFS v4 ACLs via extended attributes when copying file.

  The following `cp --preserve=xattr file1 file2` command makes a copy of the file1 file as the file2 file while making a copy of the ACL of the file1 file for the file2 file. The `cp -ar dir1 dir2` command makes a copy of the dir1 directory as the dir2 directory while making a copy of the ACL of the dir1 directory for the dir2 directory.

  > **Note** You may fail to migrate NFSv4 ACLs if the version of the rsync tool is earlier than 3.1.2.

```
[root@vbox nfs]# nfs4_getfacl file1
# file: file1
A::OWNER@:rwatTcCy
A::GROUP@:rwatcy
A::EVERYONE@:rwatcy
A::1000:rtncy
[root@vbox nfs]# cp --preserve=xattr file1 file2
```

```
[root@vbox nfs]# nfs4_getfacl file2
# file: file2
A::OWNER@:rwatTcCy
A::GROUP@:rwatcy
A::EVERYONE@:rwatcy
A::1000:rtncy
[root@vbox nfs]# cp -ar dir1 dir2
```

- Interaction between NFSv4 ACLs and file mode creation masks is supported. The modification for the ACL of an object may change the file mode creation mask of the object. The modification for the file mode creation mask of an object may change the ACL of the object.

  For example, the file mode creation mask of the file object is 0666.

  ```
  [root@vbox nfs]# ls -l file
  -rw-rw-rw-. 1 root root 0 May  3  2019 file
  [root@vbox nfs]# nfs4_getfacl file
  # file: file
  A::OWNER@:rwatTcCy
  A::GROUP@:rwatcy
  A::EVERYONE@:rwatcy
  ```

  - If you add the execute permission to the file mode creation mask by modifying the owner bits, the execute permission is also added to the ACE that includes the OWNER@ principal.

    ```
    [root@vbox nfs]# chmod u+x file
    [root@vbox nfs]# ls -l file
    -rwxrw-rw-. 1 root root 0 May  3  2019 file
    [root@vbox nfs]# nfs4_getfacl file
    # file: file
    A::OWNER@:rwaxtTcCy
    A::GROUP@:rwatcy
    A::EVERYONE@:rwatcy
    ```

  - If you add the execute permission to an ACE that includes the GROUP@ principal, the execute permission is also added to the related file mode creation mask.

    ```
    [root@vbox nfs]# nfs4_setfacl -a A::GROUP@:x file
    [root@vbox nfs]# ls -l file
    -rwxrwxrw-. 1 root root 0 May  3  2019 file
    ```

  > **Note**
  >
  > - In the interaction between ACLs and file mode creation masks, the EVERYONE@ principal is equal to the others class. When you modify the others class, the change also applies to the EVERYONE@ principal. This operation results in a slight impact on the semantics of permissions. For example, the current file mode creation mask is 177. After you run the `chmod o+r` command, all users that include the file owner and group members have the read permission. This occurs because the read permission is added to the related ACE that includes the EVERYONE@ principal. If no change is applied to the default file mode creation mask, the owner and group classes still have no read permission after you run the chmod o+r command.
  > - If no change is applied to NFSv4 ACLs, the others class of the file mode creation mask keeps the same semantics. If an NFSv4 ACL is changed, the semantics of the others class change to the semantics of the EVERYONE@ principal and the latest semantics remain. We recommend that you do not use file mode creation masks after using NFSv4 ACLs.

- Interaction between NFSv4 ACLs and POSIX ACLs is supported.

You can mount NFSv3 file systems that have NFSv4 ACLs applied. These NFSv4 ACLs will then be converted into POSIX ACLs. You can also mount NFSv4 file systems that have POSIX ACLs applied. These POSIX ACLs will then be converted into NFSv4 ACLs.

> **Note** The semantics of POSIX ACLs are different from the semantics of NFSv4 ACLs. For example, the inheritance rules that apply to POSIX ACLs do not differentiate files and directories. NFSv4 ACLs have more permissions than POSIX ACLs, which have only read, write, and execute permissions. We recommend that you use either NFSv4 ACLs or POSIX ACLs to prevent against potential issues.

For example, you configure an NFSv4 ACL for the dir0 directory. The permissions are listed as follows.

```
[root@vbox test] sudo nfs4_getfacl dir0
A::OWNER@:tTnNcCy
A::GROUP@:tncy
A::EVERYONE@:tncy
A:fdi:EVERYONE@:tncy
A:fdi:OWNER@:tTnNcCy
A:fdi:GROUP@:tncy
A:g:19064:rxtncy
A:g:19065:rwaDxtTnNcCy
A:fdig:19064:rxtncy
A:fdig:19065:rwaDxtTnNcCy
```

You configure a POSIX ACL for the dir0 directory. The permissions are listed as follows.

```
[root@vbox test] sudo getfacl dir0
user::---
group::---
group:players:r-x
group:adminis:rwx
mask::rwx
other::---
default:user::---
default:group::---
default:group:players:r-x
default:group:adminis:rwx
default:mask::rwx
default:other::---
```

For example, you configure an NFSv4 ACL for the dir0/file file. The permissions are listed as follows.

```
[root@vbox test] sudo nfs4_getfacl dir0/file
A::OWNER@:tTnNcCy
A::GROUP@:tncy
A::EVERYONE@:tncy
A:g:19064:rxtncy
A:g:19065:rwaxtTnNcCy
```

For example, you configure a POSIX ACL for the dir0/file file. The permissions are listed as follows.

```
[root@vbox test] sudo getfacl dir0/file
user::---
group::---
group:players:r-x
group:adminis:rwx
mask::rwx
other::---
```

- The number of NFSv4 ACLs is limited.

NAS supports a maximum of 100,000 ACLs that are different from one another in each file system. Each ACL contains a maximum of 500 ACEs.

> **Note** We recommend that you do not abuse ACLs and ACEs. This reduces the time and resources consumed for verifying permissions.

Features of NAS POSIX ACLs

- Permissions that are specified for the other class apply to all users.

  Everyone includes the owner, group, and users that are related to each ACE. The other class is equal to the EVERYONE@ principal of an NFSv4 ACL.

  > **Note** We recommend that you grant the least permissions to the other class in all cases.

  For example, the following ACL is configured for the *myfile* file. Although the ACE contains a user named alice who does not have the write permission, the write permission propagates to the ACE because the permission is specified for the other class.

  ```
  [root@vbox 3]# getfacl myfile
  # file: myfile
  # owner: root
  # group: root
  user::rw-
  user:alice:r--
  group::r--
  mask::r--
  other::rw-
  ```

- Permissions that are configured by ACLs will not be changed after you run the `chmod` command.

  > **Note** We recommend that you avoid modifying the file mode creation mask of a file that has a POSIX ACL applied. You can configure permissions for the file by modifying the POSIX ACL.

  i. For example, an ACE that grants the players group the read and write access to the *myfile* file.

  ```
  [root@vbox 3]# getfacl myfile
  # file: myfile
  # owner: root
  # group: root
  user::rw-
  user:player:rw-
  group::rw-
  group:players:rw-
  mask::rw-
  other::---
  ```

  ii. The `chmod g-w myfile` or `chmod u-w myfile` command does not change the permissions that are granted to the player user and the players group, which is different from the POSIX ACL standard. However, this ensures that permissions that are granted by POSIX ACLs to non-reserved users are the same after you modify permissions by using file mode creation masks. The non-reserved users include all users except for the users of the owner, group, and other classes.

```
[root@vbox 3]# getfacl myfile
# file: myfile
# owner: root
# group: root
user::r--
user:player:rw-
group::r--
group:players:rw-
mask::rw-
other::---
```

- If the execute permission is not granted to the group and other classes of an ACL, the ACL has no execute permission.

  The rule is predefined in Linux. The execute action is allowed by the backend of NAS. However, to make the execute permission in the ACL effective, you must grant the execute permission to the group or other class.

  For example, if the group and other classes do not have the execute access to the *myfile* file, the player user cannot execute the file.

  ```
  [root@vbox 3]# getfacl myfile
  # file: myfile
  # owner: root
  # group: root
  user::rw-
  user:player:r-x
  group::r--
  mask::r-x
  other::r--
  ```

  If you grant the execute permission to the group class, the execute permission also propagates to the player user.

  ```
  [root@vbox 3]# getfacl myfile
  # file: myfile
  # owner: root
  # group: root
  user::rw-
  user:player:r-x
  group::r-x
  mask::r-x
  other::r--
  ```

- If you configure inheritable NFSv4 ACLs for directories, these settings may not conform to the POSIX ACL standard when these directories reside in NFSv3 file systems.

  Inheritance rules that apply to files are different from those that apply to directories in NFSv4 ACLs. The same inheritance rules apply to both files and directories in POSIX ACLs.

  > **Note** We recommend that you apply either NFS4 ACLs or POSIX ACLs to an NFS file system to prevent against potential issues.

- File mode creation masks cannot be modified.

  The file mode creation mask of a POSIX ACL is yielded by the combination and interaction of permissions from all users and groups. The mask has no practical meaning and cannot be changed.

- You need to manage mappings between usernames or group names and user IDs (UIDs) or group IDs (GIDs) across multiple instances.

Apsara File Storage NAS NFS adopts IP security groups rather than usernames to authenticate users. When you configure POSIX ACLs, UIDs or GIDs that are included in ACEs are stored in Linux. When you print an ACL for an object in a shell, Linux automatically loads the */etc/passwd* file and converts UIDs or GIDs into actual usernames or group names. You need to manage mappings between usernames or group names and UIDs or GIDs across multiple instances. You must ensure a username or group name is mapped to its related UID or GID.

- POSIX ACLs can be printed by using extended attributes.

```
[root@vbox nfs]# getfattr -n system.posix_acl_access file
# file: file
system.posix_acl_access=0sAgAAAAEAAAD/////AgAFACAEAAAEAAAA/////xAABQD/////IAABAP////8=
```

- POSIX ACLs can be migrated by using tools such as cp.

  NAS allows you to migrate POSIX ACLs by using the cp, tar, and rsync tools. For more information, see How to preserve NFS v4 ACLs via extended attributes when copying file.

  The following `cp --preserve=xattr file1 file2` command makes a copy of the file1 file as the file2 file while making a copy of the ACL of the file1 file for the file2 file. The `cp -ar dir1 dir2` command makes a copy of the dir1 directory as the dir2 directory while making a copy of the ACL of the dir1 directory for the dir2 directory.

  > **Note** You may fail to migrate POSIX ACLs if the version of the rsync tool is earlier than 3.1.2.

```
[root@vbox nfs]# getfacl file1
user::---
user:player:r-x
group::---
mask::r-x
other::--x
[root@vbox nfs]# cp --preserve=xattr file1 file2
```

```
[root@vbox nfs]# getfacl file2
# file: file2
user::---
user:player:r-x
group::---
mask::r-x
other::--x
[root@vbox nfs]# cp -ar dir1 dir2
```

- The number of POSIX ACLs is limited.

  NAS supports a maximum of 100,000 ACLs that are different from one another in each file system. Each ACL contains a maximum of 500 ACEs.

  > **Note** We recommend that you do not abuse ACLs and ACEs. This reduces the time and resources consumed for verifying permissions.

FAQ

Why are deny ACEs not supported?

- The position of an ACE that resides in an ACL is important.

  The sequence for ACEs that reside in an NFSv4 ACL is random. A deny ACE may be placed in any position of an NFSv4 ACL. For example, an ACL contains two ACEs: A::Alice:r and D::Alice:r. The position of the ACEs determines whether the user named Alice has the write permission.

> **Note** When you configure an ACL, you must consider the position of each ACE.

- The number of ACEs in an ACL experiences a sharp increase.

  You may have difficulties to combine and deduplicate ACEs in an ACL because the sequencing for ACEs is not mandatory. The number of ACEs may increase up to tens or hundreds over a long period of time. To manage the final permissions that are produced by these ACEs, you need to check each ACE. The process to check is strenuous and time-consuming.

- The interactions between file mode creation masks and ACLs become more complex after deny ACEs are applied because deny features do not exist in file mode creation masks.
  - If deny ACEs are available, you may need to add several ACEs to an ACL when the file mode creation mask is changed. For example, if you change the file mode creation mask to -rw-rw-rw, you need to add the following ACEs to an ACL. You must add the ACEs in sequence at the beginning of the ACL.

    ```
    A::OWNER@:rw
    D::OWNER@:x
    A::GROUP@:rw
    D::GROUP@:x
    A::EVERYONE@:rw
    D::EVERYONE@:x
    ```

  - If deny ACEs are unavailable, you can sequence and deduplicate ACEs. You do not need to differentiate the EVERYONE@ principal and the other class. You can modify an ACL with ease when the file mode creation mask is changed. In such cases, you only need to find ACEs that contain the OWNER@, GROUP@, and EVERYONE@ principals and modify these ACEs as follows.

    ```
    A::OWNER@:rw
    A::GROUP@:rw
    A::EVERYONE@:rw
    ```

- Conversions between NFSv4 ACLs and POSIX ACLs are not supported in some cases.

  POSIX ACLs do not support deny ACEs. If deny ACEs are included in an NFSv4 ACL, you cannot convert the ACL into a POSIX ACL.

7.1.10.3. Use POSIX ACLs to control access

This topic describes how to configure Portable Operating System Interface (POSIX) access control lists (ACLs). You can use POSIX ACLs to control access to files and directories that reside in an NFSv3 file system.

Prerequisites

An NFSv3 file system is mounted. For more information, see Mount an NFS file system.

Commands

Before you configure POSIX ACLs, we recommend that you familiarize yourself with the related commands.

Command	Description
getfacl <filename>	Shows the ACL that applies to the specified file.
setfacl -m g::w <filename>	Grants the owing group the write access.
setfacl -m u:player:w <filename>	Grants the player user the write access.
setfacl -m g:players:rwx <filename>	Grants the players group the read, write, and execute access.

Command	Description
setfacl -x g:players <filename>	Removes permissions from the players group
getfacl file1 | setfacl --set-file=- file2	Copies the ACL for the *file1* file to the *file2* file.
setfacl -b file1	Removes all extended ACEs from the *file1* file. The base ACEs of the owner, group, and others are retained.
setfacl -k file1	Removes all default ACEs from the *file1* file.
nfs4_setfacl -R -m g:players:rw dir	Grants the players group the read and write access to files and subdirectories in the *dir* directory.
setfacl -d -m g:players:rw dir1	Grants the players group the read and write access to the newly created files and subdirectories in the *dir1* directory.

Procedure

To control access to files and directories by configuring NFS ACLs, follow these steps.

1. Create users and groups.

   In this example, the following users are created: player, admini, and anonym. The following groups are created: players and adminis. The player user is added to the players group and the admini user is added to the adminis group.

   ```
   sudo useradd player
   sudo groupadd players
   sudo usermod -g players player
   sudo useradd admini
   sudo groupadd adminis
   sudo usermod -g adminis admini
   sudo useradd anonym
   ```

2. Configure POSIX ACLs to control access to files and directories.

   Use the following commands to complete the operations: create a directory named *dir0* and grant the players group the read-only access, the adminis group the read, write, and execute permissions, and the others class no access to all the files in the *dir0* directory.

   ```
   sudo umask 777
   sudo mkdir dir0
   sudo setfacl -m g:players:r-x dir0
   sudo setfacl -m g:adminis:rwx dir0
   sudo setfacl -m u::--- dir0
   sudo setfacl -m g::--x dir0
   sudo setfacl -m o::--- dir0
   sudo setfacl -d -m g:players:r-x dir0
   sudo setfacl -d -m g:adminis:rwx dir0
   sudo setfacl -d -m u::--- dir0
   sudo setfacl -d -m g::--x dir0
   sudo setfacl -d -m o::--- dir0
   ```

   Use the `sudo getfacl dir0` command to verify the result after the configuration is complete.

```
# file: dir0
# owner: root
# group: root
user::---
group::--x
group:players:r-x
group:adminis:rwx
mask::rwx
other::---
default:user::---
default:group::--x
default:group:players:r-x
default:group:adminis:rwx
default:mask::rwx
default:other::---
```

3. Verify the ACL configuration.
   i. Verify that the admini user has read and write access to the dir0/file file.

```
[root@vbox test] sudo su admini -c 'touch dir0/file'
[root@vbox test] sudo su admini -c 'echo 123 > dir0/file'
```

   ii. Use the following command to verify the read-only access of the player user.

```
[root@vbox test] sudo su player -c 'touch dir0/file'
touch: cannot touch ‘dir0/file’: Permission denied
[root@vbox test] sudo su player -c 'cat dir0/file'
123
[root@vbox test] sudo su player -c 'echo 456 >> dir0/file'
bash: dir0/file: Permission denied
[root@vbox test] sudo su player -c 'getfacl dir0/file'
# file: dir0/file
# owner: admini
# group: adminis
user::---
group::---
group:players:r-x
group:adminis:rwx
mask::rwx
other::---
```

   iii. Verify that the anonym user does not have access to the dir0/file file.

```
[root@vbox test] sudo su anonym -c 'ls dir0'
ls: cannot open directory dir0: Permission denied
[root@vbox test] sudo su anonym -c 'cat dir0/file'
cat: dir0/file: Permission denied
[root@vbox test] sudo su anonym -c 'getfacl dir0/file'
getfacl: dir0/file: Permission denied
```

Related operations

If you want to remove user permissions, use the following method.

When you use NFSv4 ACLs, we recommend that you sort each user into different groups. This allows you to configure permissions for a group rather than a separate user. To disable access to an object from a user, you can remove the user from a group that has access to the object. For example, the following commands remove the admini user from the adminis group and add the user to the adminis2 group.

```
[root@vbox test] sudo groupadd adminis2
[root@vbox test] sudo usermod -g adminis2 admini
[root@vbox test] id admini
uid=1057(admini) gid=1057(admini) groups=1061(adminis2)
[root@vbox test] sudo su admini -c 'ls dir0'
ls: cannot open directory dir0: Permission denied
[root@vbox test] sudo su admini -c 'cat dir0/file'
cat: dir0/file: Permission denied
[root@vbox test] sudo su admini -c 'getfacl dir0/file'
getfacl: dir0/file: Permission denied
```

7.1.10.4. Use NFSv4 ACLs to control access

This topic describes how to configure NFSv4 access control lists (ACLs) and apply these ACLs to NFSv4 file systems to control access to files and directories.

Prerequisites

An NFSv4 file system is mounted. For more information, see Mount an NFS file system.

Context

You can mount an NFSv4 file system on an Elastic Compute Service (ECS) instance that runs Linux and install the Linux-specific nfs4-acl-tools tool on the instance. You can use the standard nfs4_getfacl and nfs4_setfacl tools to configure NFSv4 ACLs after the installation is completed.

Commands

Before you configure NFSv4 ACLs, we recommend that you familiarize yourself with the related commands.

Command	Description
`nfs4_getfacl <filename>`	Views the access permissions for the specified file.
`nfs4_setfacl -a A::GROUP@:W <filename>`	Adds an access control entry (ACE) that grants the GROUP@ principal the write access to the specified file.
`nfs4_setfacl -a A::1000:W <filename>`	Adds an ACE that grants a user principal named 1000 the write access to the specified file.
`nfs4_setfacl -a A:g:10001:W <filename>`	Adds an ACE that grants a group principal named 10001 the write access to the specified file.
`nfs4_setfacl -e <filename>`	Edits an ACL in an interactive mode.
`nfs4_getfacl <filename> > saved_acl.txt`	Saves a list of permissions for the specified file as a TXT file.
`nfs4_setfacl -S saved_acl.txt <filename>`	Configures permissions for the specified file by using a TXT file that includes a list of ready-made permissions.
`nfs4_setfacl -m A::1001:rwaxTNcCy A::1001:rxtcy file1`	Modifies the permission of an ACE that applies to the *file1* file.
`nfs4_getfacl file1 | nfs4_setfacl -S - file2`	Copies the permissions for the *file1* file to the *file2* file.

Command	Description
`nfs4_getfacl file1 | grep @ | nfs4_setfacl -S - file1`	Deletes all ACEs that apply to the *file1* file except for ACEs that include the following principals: OWNER@, GROUP@, and EVERYONE@.
`nfs4_setfacl -R -a A:g:10001:rW dir`	Adds an ACE that grants a group principal named 10001 the read and write access to files and subdirectories in the *dir* directory.
`find dir -type f -exec sh -c 'for ace in $(nfs4_getfacl \{} | grep "^A.*\:1005\:"); do nfs4_setfacl -x $ace \{}; done' \;`	Deletes ACEs that grant a user principal named 1005 access to files in the *dir* directory.
`nfs4_setfacl -a A:fdg:10001:rW dir1`	Adds an ACE that grants a group principal named 10001 the read and write access to new files and subdirectories in the *dir1* directory.
`nfs4_setfacl -a A:fg:10001:rx dir1`	Adds an ACE that grants a group principal named 10001 the read, write, and execute access to all newly created files in the *dir1* directory.

Procedure

To control access to files and directories by configuring NFSv4 ACLs, follow these steps.

1. Create users and groups.

   In this example, the following users are created: player, admini, and anonym. The following groups are created: players and adminis. The player user is added to the players group and the admini user is added to the adminis group.

   ```
   sudo useradd player
   sudo groupadd players
   sudo usermod -g players player
   sudo useradd admini
   sudo groupadd adminis
   sudo usermod -g adminis admini
   sudo useradd anonym
   ```

2. Install the related tools to configure NFSv4 ACLs.

   If you have installed these tools, skip this step.

   ```
   sudo yum -y install nfs4-acl-tools
   ```

3. Obtain the group IDs of the players and adminis groups.

   Open the */etc/group* file. The group IDs of the players and adminis groups are displayed as follows:

   ```
   players:x:19064:player
   adminis:x:19065:admini
   ```

4. Configure NFSv4 ACLs for files and directories.

   Use the following commands to complete the operations: create a directory named *dir0* and add ACEs that grant the players group the read-only access, the adminis group the read, write, and execute access, and other users no access to all the files in the *dir0* directory.

```
sudo umask 777
sudo mkdir dir0
sudo nfs4_setfacl -a A:fdg:19064:RX dir0
sudo nfs4_setfacl -a A:fdg:19065:RWX dir0
sudo nfs4_setfacl -a A:fdg:OWNER@: dir0
sudo nfs4_setfacl -a A:fdg:GROUP@: dir0
sudo nfs4_setfacl -a A:fdg:EVERYONE@: dir0
```

Use the `sudo nfs4_getfacl dir0` command to verify the configuration.

```
A::OWNER@:tTnNcCy
A::GROUP@:tncy
A::EVERYONE@:tncy
A:fdi:EVERYONE@:tncy
A:fdi:OWNER@:tTnNcCy
A:fdi:GROUP@:tncy
A:g:19064:rxtncy
A:g:19065:rwaDxtTnNcCy
A:fdig:19064:rxtncy
A:fdig:19065:rwaDxtTnNcCy
```

5. Verify the configuration of the ACL.
   i. Use the following commands to verify the read and write access of the admini user.

   ```
   [root@vbox test] sudo su admini -c 'touch dir0/file'
   [root@vbox test] sudo su admini -c 'echo 123 > dir0/file'
   ```

   ii. Use the following command to verify the read-only access of the player user.

   ```
   [root@vbox test] sudo su player -c 'touch dir0/file'
   touch: cannot touch ‘dir0/file’: Permission denied
   [root@vbox test] sudo su player -c 'echo 456 >> dir0/file'
   bash: dir0/file: Permission denied
   [root@vbox test] sudo su player -c 'cat dir0/file'
   123
   [root@vbox test] sudo su player -c 'nfs4_getfacl dir0/file'
   A::OWNER@:tTnNcCy
   A::GROUP@:tncy
   A::EVERYONE@:tncy
   A:g:19064:rxtncy
   A:g:19065:rwaxtTnNcCy
   ```

   iii. Use the following command to verify that the anonym user does not have access to the /dir0/file file.

   ```
   [root@vbox test] sudo su anonym -c 'ls dir0'
   ls: cannot open directory dir0: Permission denied
   [root@vbox test] sudo su anonym -c 'cat dir0/file'
   cat: dir0/file: Permission denied
   [root@vbox test] sudo su anonym -c 'nfs4_getfacl dir0/file'
   Invalid filename: di
   ```

Related operations

If you want to remove user permissions, use the following method.

When you use NFSv4 ACLs, we recommend that you sort each user into different groups. This allows you to configure permissions only for a group rather than a separate user. To disable access to an object from a user, you can remove the user from a group that has access to the object. For example, use the following commands to remove the admini user from the adminis group and add the user to the adminis2 group:

```
[root@vbox test] sudo groupadd adminis2
[root@vbox test] sudo usermod -g adminis2 admini
[root@vbox test] id admini
uid=1057(admini) gid=1057(admini) groups=1054(adminis2)
[root@vbox test] sudo su admini -c 'ls dir0'
ls: cannot open directory dir0: Permission denied
[root@vbox test] sudo su admini -c 'cat dir0/file'
cat: dir0/file: Permission denied
[root@vbox test] sudo su admini -c 'nfs4_getfacl dir0/file'
Invalid filename: dir0/file
```

8.Tablestore

8.1. User Guide

8.1.1. What is Tablestore?

Tablestore is a NoSQL database service independently developed by Alibaba Cloud. Tablestore is a proprietary software program that is certified by the relevant authorities in China. Tablestore is built on the Apsara system of Alibaba Cloud, and can store large amounts of structured data and allow real-time access to the data.

Tablestore provides the following features:

- Offers schema-free data storage. You do not need to define attribute columns before you use them, or perform table-level changes to add or delete attribute columns. You can set the time to live (TTL) parameter for a table to manage the lifecycle of data. Expired data is deleted from the table.
- Uses the multi-node cluster architecture. Each management node on the platform implements a high availability mechanism. Therefore, the failure of a service node within a cluster does not affect the overall operating of businesses.
- Adopts the triplicate technology to keep three copies of data across different servers in different racks. A cluster can support single storage type instances (SSD only) or mixed storage type instances (SSD and SATA) to meet different budget and performance requirements.
- Adopts a fully redundant architecture to prevent single points of failure (SPOFs). The support for smooth online upgrades, hot cluster upgrades, and automatic data migration enables you to dynamically add or remove nodes for maintenance without incurring service interruptions. The concurrent read/write throughput and storage capacity can be linearly scaled. Each cluster can have no less than 500 hosts.
- Supports highly concurrent read/write operations. Concurrent read/write capabilities can be scaled out as the number of hosts increases. The read/write performance is indirectly related to the amount of data in a single table.
- Supports identity authentication and multi-tenancy. Comprehensive access control and isolation mechanisms are provided to safeguard your data. VPC and access over HTTPS are supported. Provides multiple authentication and authorization mechanisms so that you can define access permissions on individual tables and operations.

8.1.2. Precautions

Before you use Tablestore, you need to take note of the following precautions and limits.

The following table describes the limits for Tablestore. A part of the limits indicate the maximum allowable values rather than the suggested values. To ensure better performance, set the table scheme and data size in a single row based on actual conditions, and adjust the following configurations.

Item	Limit	Description
The number of instances under an Apsara Stack tenant account	1024	To raise the limit, contact the technical support personnel.
The number of tables in an instance	1024	To raise the limit, contact the technical support personnel.
The length of an instance name	3 to 16 bytes	The instance name can contain uppercase and lowercase letters, digits, and hyphens (-). It must start with a letter and cannot end with a hyphen (-).

Item	Limit	Description
The length of a table name	1 to 255 bytes	The table name can contain uppercase and lowercase letters, digits, and underscores (_). It must start with a letter or underscore (_).
The length of a column name	1 to 255 bytes	The column name can contain uppercase and lowercase letters, digits, and underscores (_). It must start with a letter or underscore (_).
The number of columns in a primary key	1 to 4	A primary key can contain one to four primary key columns.
The size of the value in a string type primary key column	1 KB	The size of the value in a STRING primary key column cannot exceed 1 KB.
The size of the value in a STRING attribute column	2 MB	The size of the value in a STRING attribute column cannot exceed 2 MB.
The size of the value in a BINARY primary key column	1 KB	The size of the value in a BINARY primary key column cannot exceed 1 KB.
The size of the value in a BINARY attribute column	2 MB	The size of the value in a BINARY attribute column cannot exceed 2 MB.
The number of attribute columns in a single row	Unlimited	A single row can contain an unlimited number of attribute columns.
The number of attribute columns written by one request	1,024	During a PutRow, UpdateRow, or BatchWriteRow operation, the number of attribute columns written in a row cannot exceed 1,024.
The data size of a row	Unlimited	The total size of all column names and column values for a row is unlimited.

8.1.3. Quick start

8.1.3.1. Log on to the Tablestore console

This topic describes how to log on to the Tablestore console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

> **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
>
> - Uppercase or lowercase letters
> - Digits
> - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products > Tablestore**.

8.1.3.2. Create instances

An instance is a logical entity in Tablestore and is used to manage tables. An instance is the basic unit of the resource management system of Tablestore. Tablestore implements application access control and resource measurement at the instance level. This topic describes how to create an instance.

Procedure

1. Log on to the Tablestore console.
2. On the **Overview** tab, click **Create Instance**.

> **Note** You can create different instances to manage the tables for different businesses, or create different instances for development, testing, and production environments of the same business. By default, Tablestore allows you to create up to 1,024 instances and up to 1,024 tables in each instance under an Apsara Stack tenant account.

3. On the **Create Tablestoree Instance** page, configure the following parameters.

Parameter	Description
Region	Select a region from the drop-down list for the instance.
Organization	Select an organization from the drop-down list for the instance.
Resource Set	Select a resource set from the drop-down list for the instance.
Instance Name	Enter a name for the instance. Instance naming conventions: The name must be 3 to 16 characters in length and can contain only letters, digits, and hyphens (-). It must start with a letter and cannot start with case insensitive string `ali` or `ots`.
Description	Enter a description for the instance.
Instance Type	Select an instance type from the drop-down list for the instance. Tablestore provides high-performance instances and capacity instances. The instance types vary based on the type of cluster you deploy.

4. Click **Submit**.
5. In the **Submitted.** dialog box, click **Back to Console**.

   On the **Overview** tab, you can view the created instance.

   After the instance is created, you can perform the following operations on the instance:

- Click the instance name or click **Manage Instance** in the Actions column. On the **Instance Management** page, click each tab to perform various operations.
  - On the **Instance Details** tab, you can view the Instance Access URL, Basic Information, and Tables sections.
  - On the **Instance Monitoring** tab, you can view monitoring data by using time ranges, metric categories, and operations.
  - On the **Network Management** tab, you can bind or unbind VPCs and view the list of VPCs.
- Click **Release** in the **Actions** column to release an instance.

  > **Notice**
  > - Before you release an instance, ensure that all tables are deleted, and VPCs are unbound from instances.
  > - To create an instance when you release an existing instance, ensure that the name of the instance you want to create is different from that of the existing instance to avoid conflicts.

8.1.3.3. Create tables

This topic describes how to create a table in the Tablestore console.

Procedure

1.
2. On the **Overview** page, click the name of the required instance or click **Manage Instance** in the Actions column corresponding to the instance.
3. On the **Instance Details** tab, click **Create Table**.

   > **Note** You can create a maximum of 1,024 tables in each instance.

4. In the **Create Table** dialog box, set **Table Name** and **Primary Key**.

   The following table describes the parameters you can configure.

Parameter	Description
Table Name	The name of the table. This name is used to uniquely identify a table in an instance. The name must be 1 to 255 bytes in length and can contain letters, digits, and underscores (_). The name must start with a letter or an underscore (_).

Parameter	Description
Primary Key	One or more primary key columns in the table that uniquely identify each record in the table. Enter a primary key name and select a data type. Click **Add a Primary Key** to add a primary key column. You can add one to four primary key columns. By default, the first primary key column is the partition key. The configurations and order of primary key columns cannot be modified after the table is created. **Note** ◦ In Tablestore, only a primary key column can be used as an auto-increment primary key column. Partition keys cannot be used as auto-increment primary key columns. ◦ After a primary key column is set to an auto-increment primary key column, Tablestore automatically generates a value for the auto-increment primary key column when you write a row of data. You do not need to specify a value for the auto-increment primary key column. The values of auto-increment primary key columns are incremental and unique within the rows that share the same partition key. ◦ Naming conventions of primary key columns: The name must be 1 to 255 bytes in length and can contain letters, digits, and underscores (_). The name must start with a letter or underscore (_) ◦ Data types supported by primary key columns are **String**, **Integer**, and **Binary**.

5. Optional. Configure advanced parameters.

   If you need to configure parameters such as Time to Live and Max Versions, perform this operation.

   i. Turn on **Advanced Settings**.

ii. Configure advanced parameters.

The following table describes the advanced parameters you can configure.

Parameter	Description
Time to Live	The period for which data in the table can be retained. When the retention period exceeds the Time to Live (TTL) value, the system deletes the expired data. The minimum TTL value is 86,400 seconds (one day). A value of -1 indicates that data never expires.
Max Versions	The maximum number of versions of data that can be retained for an attribute column. When the versions of data in an attribute column exceed the Max Versions value, the system deletes the earliest versions of data to keep the maximum number of versions equal to the Max Versions value. Valid values: 1 to 10.
Max Version Offset	The difference between the version number and the data written time must be within the value of Max Version Offset. Otherwise, an error occurs when the data is written. Unit: seconds. The valid version range for attribute columns is calculated based on the following formula: Valid version range = **[Data written time - Max version offset value, Data written time + Max version offset value)**.
Reserved Read Throughput	You can set this parameter only for high-performance instances. The read and write throughput that is allocated and reserved for the table. Valid values: integers from 0 to 5000. When the specified reserved read and write throughput is **0**, Tablestore does not reserve related resources for the table.
Reserved Write Throughput	

6. Optional. Create secondary indexes.

If you need to create secondary indexes, perform this operation.

i. Turn on **Create Secondary Index**.

ii. Click the **+ Add** button in the Pre-defined Column section. Enter the name of the pre-defined column and select a data type from the drop-down list.

- This operation is performed to create a predefined column for the base table. Tablestore uses a schema-free model. You can write an unlimited number of columns to a row and do not need to specify a fixed number of predefined columns in a schema. When you create a table, you can also predefine columns and specify their data types.
- You can add up to 14 predefined columns. To deleted the predefined column you add, click the icon on the left of the corresponding predefined column.
- The name of a predefined column must be 1 to 255 bytes in length and can contain letters, digits, and underscores (_). The name must start with a letter or underscore (_).
- The data types of predefined columns include STRING, INTEGER, BINARY, FLOAT, and BOOLEAN.

iii. Click **Add Secondary Index**. Enter Index Name and set Primary Key and Pre-defined Column for the index table.

- The name of an index table must be 1 to 255 bytes in length and can contain letters, digits, and underscores (_). The name must start with a letter or underscore (_).
- You can set the primary key of the index table to the primary key or predefined columns of the base table.
- Pre-defined Column is optional. You can set the predefined columns of the index table to only the predefined columns of the base table.

7. Click **OK**.

   After a table is created, you can view the table in the **Table List** section. If the created table is not displayed in the list of tables, click the icon to refresh the list of tables.

   After a table is created, you can perform the following operations on the table:

   - Click the name of the table or click Details in the Actions column. On the **Manage Table** page, you can perform the following operations:
     - On the **Details** tab, you can view the description of the table and the primary key columns list, and modify the attributes of the table.
     - On the **Data Editor** tab, you can insert or update data, query data, view data details, and delete multiple data at a time.
   - Click the icon in the Actions column corresponding to a table and choose **Delete** from the shortcut menu. Click OK in the Delete Table dialog box. The table is deleted.

     Notice If you delete a table, the table and the data in the table are permanently deleted from Tablestore and cannot be recovered. Exercise caution when you perform this operation.

8.1.3.4. Read and write data in the console

After a table is created, you can read data from and write data to the table in the console.

Add data

1.
2. On the **Overview** page, click the name of the required instance or click **Manage Instance** in the Actions column corresponding to the instance.
3. In the **Table List** section of the **Instance Details** tab, click the name of the target table and click the **Data Editor** tab. You can also click **Data Editor** in the Actions column.
4. On the **Data Editor** tab, click **Insert**.
5. In the **Insert** dialog box that appears, set **Primary Key Value**. Click **Add Column**. Set **Name**, **Type**, **Value**, and **Version**.

   By default, **System Time** is selected, indicating that the current system time is used as the version number of the data. You can also clear **System Time** and enter the version number of the data.
6. Click **OK**.

   Rows that contain the written data are displayed on the **Data Editor** tab.

Update data

You can update data in the attribute columns of a row.

1.
2. On the **Overview** page, click the name of the required instance or click **Manage Instance** in the Actions column corresponding to the instance.
3. In the **Table List** section of the **Instance Details** tab, click the name of the target table and click the **Data Editor** tab. You can also click **Data Editor** in the Actions column.
4. On the **Data Editor** tab, select the row of data to update. Click **Update**.
5. In the **Update** dialog box that appears, modify the type and value for the primary key, add or remove attribute columns, and update or delete data in attribute columns.
   - You can click **+Add Column** to add an attribute column. You can also click the icon to delete an attribute column.
   - If you select **Update**, you can modify data in attribute columns. If you select **Delete**, select the number of version to delete. If you select **Delete All**, all versions of the data are deleted.
6. Click **OK**.

Query data

In the Tablestore console, you can query data in a single row (GetRow) or query data within a specified range (RangeQuery).

To query data in a single row, perform the following operations:

1.
2. On the **Overview** page, click the name of the required instance or click **Manage Instance** in the Actions column corresponding to the instance.
3. In the **Table List** section of the **Instance Details** tab, click the name of the target table and click the **Data Editor** tab. You can also click **Data Editor** in the Actions column.
4. On the **Data Editor** tab, click **Search**.
5. Set filter conditions.
   i. In the **Search** dialog box, Set Modes to **GetRow**.
   ii. By default, the system returns all columns. To return specified attribute columns, turn off **All Columns**. Enter the names of the attribute columns to return. Separate the names of the attribute columns with commas (,).
   iii. Set **Primary Key Value**.
   The integrity and accuracy of the primary key value affect the query results.
   iv. Set **Count of Versions** to specify the maximum number of versions to return.
6. Click **OK**.
   Data that meets the filter conditions is displayed on the **Data Editor** tab.

To perform range query, perform the following steps:

1.
2. On the **Overview** page, click the name of the required instance or click **Manage Instance** in the Actions column corresponding to the instance.
3. In the **Table List** section of the **Instance Details** tab, click the name of the target table and click the **Data Editor** tab. You can also click **Data Editor** in the Actions column.
4. On the **Data Editor** tab, click **Search**.
5. Set filter conditions.
   i. In the **Search** dialog box, Set Modes to **Range Search**.

ii. By default, the system returns all columns. To return specified attribute columns, turn off **All Columns**. Enter the names of the attribute columns to return. Separate the names of the attribute columns with commas (,).

iii. Set Start Primary Key Column and End Primary Key Column.

You can set Start Primary Key Column to **Min Value** or **Custom** and End Primary Key Column to **Max Value** or **Custom**. If you select **Custom**, enter a custom value.

> Note
>
> - The value in the first primary key column takes priority when the range query mode is used. When the minimum and maximum values for the first primary key column are the same, the system uses the value in the second primary key column to perform the query. The query rules for the subsequent primary keys are the same as those for the first two primary keys.
> - The Custom range is a left-open and right-closed interval.

iv. Set **Count of Versions** to specify the maximum number of versions to return.

v. Set Sequence to **Forward Search** or **Backward Search**.

6. Click **OK**.

Data that meets the filter conditions is displayed based on the specified order on the **Data Editor** tab.

Delete data

You can delete data you no longer need.

1.
2. On the **Overview** page, click the name of the required instance or click **Manage Instance** in the Actions column corresponding to the instance.
3. In the **Table List** section of the **Instance Details** tab, click the name of the target table and click the **Data Editor** tab. You can also click **Data Editor** in the Actions column.
4. On the **Data Editor** tab, select the row of data you want to delete. Click **Delete**.
5. In the **Delete** message that appears, click **OK**.

8.1.3.5. Bind a VPC to a Tablestore instance

After you bind a VPC to a Tablestore instance, you can access the Tablestore instance from the ECS instances in the VPC in the same region.

Prerequisites

- A VPC that is within the same region as the Tablestore instance is created.
- After the VPC is created, create an ECS instance in the VPC.

Procedure

1. Log on to the Tablestore console.
2. On the **Overview** page, click the name of the required instance or click **Manage Instance** in the Actions column corresponding to the instance.
3. Click the **Network Management** tab.
4. On the **Network Management** tab, click **Bind VPC**.
5. In the **Bind VPC** dialog box, select a VPC and switch, enter **Instance VPC Name**.

The name of a VPC can contain only letters and digits and must start with a letter. The name must be 3 to 16 bytes in length.

6. Click **OK**.

   After the VPC is bound to the instance, you can view the information of the VPC in the **VPC List** on the **Network Management** tab. You can use the VPC address to access the Tablestore instance from the ECS instances in the VPC.

   After you bind a VPC, you can perform the following operations:

   - Click **VPC Instance List** in the Actions column to view the VPC instances list, which contains the instance name, instance VPC name, and VPC domain name.
   - Click **Unbind** in the Actions column to unbind the VPC from the instance. After the VPC is unbound, the ECS instance in the VPC can no longer access the Tablestore instance by using the VPC address. To access the Tablestore instance from the ECS instance, you must bind the VPC to the Tablestore instance again.

9.ApsaraDB RDS for MySQL

9.1. User Guide (RDS for MySQL)

9.1.1. What is ApsaraDB RDS?

ApsaraDB Relational Database Service (RDS) is a stable, reliable, and scalable online database service. Based on the distributed file system and high-performance storage, ApsaraDB RDS provides a set of solutions for disaster recovery, backup, restoration, monitoring, and migration.

ApsaraDB RDS supports four storage engines, which are MySQL, SQL Server, PolarDB, and PostgreSQL. You can create database instances based on these storage engines to meet your business requirements.

RDS MySQL

Originally based on a branch of MySQL, ApsaraDB RDS for MySQL provides excellent performance. It is a tried and tested solution that handled the high-volume concurrent traffic during Double 11. ApsaraDB RDS for MySQL provides basic features, such as whitelist configuration, backup and restoration, Transparent Data Encryption (TDE), data migration, and management for instances, accounts, and databases. ApsaraDB RDS for MySQL also provides the following advanced features:

- **Read-only instance:** In scenarios where ApsaraDB RDS handles a small number of write requests but a large number of read requests, you can create read-only instances to scale up the reading capability and increase the application throughput.
- **Read/write splitting:** The read/write splitting feature provides a read/write splitting endpoint. This endpoint enables an automatic link for the primary instance and all its read-only instances. An application can connect to the read/write splitting endpoint to read and write data. Write requests are distributed to the primary instance and read requests are distributed to read-only instances based on their weights. To scale up the reading capability of the system, you can add more read-only instances.

9.1.2. Log on to the ApsaraDB RDS console

This topic describes how to log on to the ApsaraDB RDS console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

> **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
>
> - Uppercase or lowercase letters
> - Digits
> - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products** > Database Services > **ApsaraDB RDS**.

9.1.3. Quick start

9.1.3.1. Limits

To ensure instance stability and security, ApsaraDB RDS for MySQL has some service limits, as listed in the following table.

Operation	Description
Instance parameters	Instance parameters can be modified by using the ApsaraDB RDS console or API operations. Due to security and stability considerations, only specific parameters can be modified.
Root permissions of databases	The root or system administrator permissions are not provided.
Database backup	• Logical backup can be performed by using the command line interface (CLI) or graphical user interface (GUI). • Physical backup can be performed only by using the ApsaraDB RDS console or API operations.
Database restoration	• Logical restoration can be performed by using the CLI or GUI. • Physical restoration can be performed only by using the ApsaraDB RDS console or API operations.
ApsaraDB RDS for MySQL storage engine	Only InnoDB is supported. • To ensure performance and security, we recommend that you use the InnoDB storage engine. • The TokuDB engine is not supported. Percona no longer provides support for TokuDB, which leads to bugs that cannot be fixed and business losses in extreme cases. • The MyISAM engine is not supported. Due to the inherent shortcomings of the MyISAM engine, some data may be lost. Only some existing instances use the MyISAM engine. MyISAM engine tables in newly created instances are automatically converted to InnoDB engine tables. • The Memory engine is not supported. Newly created Memory tables are automatically converted into InnoDB tables.
Database replication	ApsaraDB RDS for MySQL provides dual-node clusters based on a primary/secondary replication architecture. The secondary instances in this replication architecture are hidden and cannot be accessed directly.

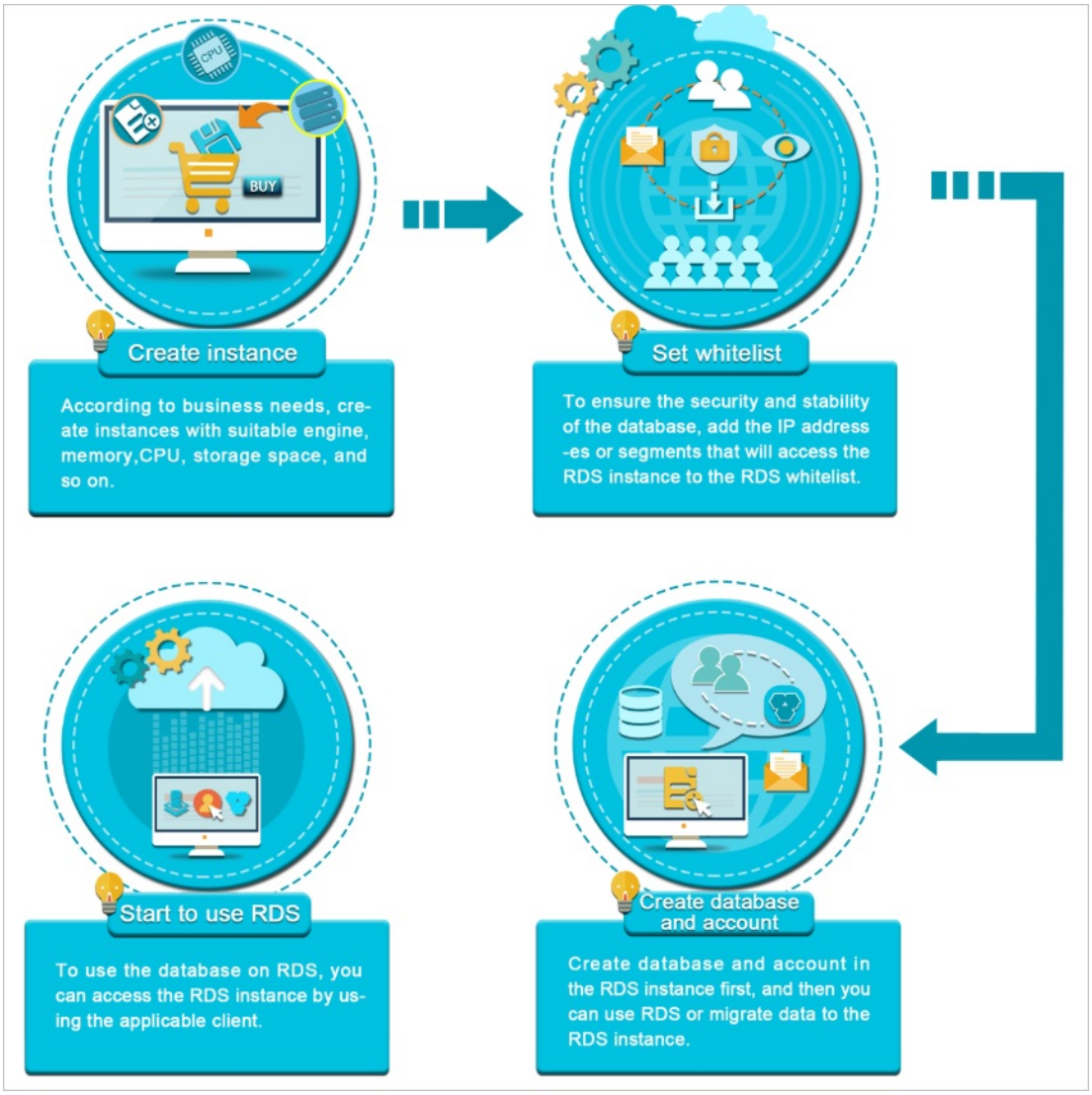
Operation	Description
Instance restart	Instances must be restarted by using the ApsaraDB RDS console or API operations.
Account and database management	ApsaraDB RDS for MySQL manages accounts and databases by using the ApsaraDB RDS console. ApsaraDB RDS for MySQL also allows you to create a privileged account to manage users, passwords, and databases.
Standard account	• Authorization is not allowed. • The ApsaraDB RDS console allows you to manage accounts and databases. • Instances that support standard accounts also support privileged accounts.
Privileged account	• Authorization is allowed to standard accounts. • The ApsaraDB RDS console does not provide interfaces to manage accounts or databases. These operations can be performed only by using code or DMS. • The privileged account cannot be reverted to a standard account.

9.1.3.2. Procedure

ApsaraDB RDS quick start covers the following operations: creating an instance, configuring a whitelist, creating a database, creating an account, and connecting to the instance. This topic describes how to use ApsaraDB RDS and provides all the necessary information to create an ApsaraDB RDS instance. ApsaraDB RDS for MySQL is used in the example.

Typically, after an instance is created, you must perform several operations to make the instance ready for use, as shown in Quick start flowchart.

Quick start flowchart



- Create an instance

  An instance is a virtual database server on which you can create and manage multiple databases.

- Configure a whitelist

  After you create an ApsaraDB RDS instance, you must configure its whitelist to allow access from external devices.

  Whitelists make your ApsaraDB RDS instance more secure. We recommend that you maintain whitelists on a regular basis. The whitelist configuration process does not affect the normal operations of the ApsaraDB RDS instance.

- Create a database and Create an account

  Before you use a database, you must first create the database and an account in the ApsaraDB RDS instance.

- Connect to an ApsaraDB RDS for MySQL instance

  After you create an ApsaraDB RDS instance, configure a whitelist, and create a database and an account, you can connect to the instance by using a database client.

9.1.3.3. Create an instance

This topic describes how to create one or more instances in the ApsaraDB RDS console.

Prerequisites

An Apsara Stack tenant account is obtained.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	**Organization**	The organization to which the instance belong.
	Resource Set	The resource set to which the instance belong.
Region	**Region**	The region in which you want to create the instance. Services in different regions cannot communicate over an internal network. After the instance is created, the region cannot be changed.
	Zone of Primary Node	The zone where the primary instance is deployed.
	Deployment Method	Specifies whether to deploy the primary and secondary instances in separate zones. ApsaraDB RDS supports **Multi-zone Deployment** and **Single-zone Deployment**. If you select **Multi-zone Deployment**, you must configure **Zone of Secondary Node**.
	Zone of Secondary Node	The zone where the secondary instance is deployed. This parameter is available only when **Deployment Method** is set to **Multi-zone Deployment**. **Note** If you select the same zone for both the primary and secondary instances, the deployment is equivalent to single-zone deployment.
	Quantity	The number of ApsaraDB RDS instances that you want to create. Default value: 1.
	Instance Name	The name of the instance. ○ The name must be 2 to 64 characters in length. ○ The name must start with a letter. ○ The name can contain letters, digits, and the following special characters: _ - : ○ The name cannot start with http:// or https://.

Section	Parameter	Description
Specifications	Network Type	The connection type of the instance. ApsaraDB RDS instances support the following connection types: ○ **Internet Connection**: ApsaraDB RDS instances of this connection type can be connected over the Internet. ○ **Internal Network**: ApsaraDB RDS instances of this connection type can be connected over an internal network. **Note** The value of this parameter cannot be changed after the instance is created. Proceed with caution.
	Database Engine	The database engine of the instance. Set the value to **MySQL**.
	Engine Version	The version of the database engine. Valid values: ○ 5.6 ○ 5.7
	Edition	The edition of the instance. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Type	The storage type of the instance. Local and standard SSDs are supported. **Note** Standard SSDs are supported only for MySQL 5.7 instances on High-availability Edition.
	Encrypted	Specifies whether to encrypt the standard SSD. This parameter is available only when **Storage Type** is set to **Standard SSD**. If you select Encrypted, you must specify the **Key** parameter. If you do not have a key, you must first create one in Key Management Service (KMS). For more information, see Create a CMK in *KMS User Guide*.
	Key	The key used to encrypt the standard SSD. This parameter is available only when you select **Encrypted**.
	Instance Type	The instance type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Capacity	The storage capacity of the instance, which includes the space to store data, system files, binlog files, and transaction files. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.

Section	Parameter	Description
Network	Network Type	The network type of the instance. ApsaraDB RDS instances support the following network types: ○ **Classic Network**: Cloud services in the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. **Note** Instances that use standard SSDs cannot be deployed in the classic network. ○ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Alibaba Cloud. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for enhanced security.
	VPC	The VPC in which you want to create the instance. **Note** You must specify this parameter when **Network Type** is set to **VPC**.
	vSwitch	The vSwitch in the VPC. **Note** You must specify this parameter when **Network Type** is set to **VPC**.
	IP Address Whitelist	The IP addresses that are allowed to connect to the ApsaraDB RDS instance.

4. After you configure the preceding parameters, click **Submit**.

9.1.3.4. Initialization settings

9.1.3.4.1. Configure a whitelist

To ensure database security and reliability, you must modify the whitelist of an ApsaraDB RDS instance before you enable the instance. You must add the IP addresses or CIDR blocks that are used for database access to the whitelist.

Context

The whitelist improves the access security of your ApsaraDB RDS instance. We recommend that you maintain the whitelist on a regular basis. The whitelist configuration process does not affect the normal operations of the ApsaraDB RDS instance.

To configure a whitelist, perform the following operations:

- Configure a whitelist: Add IP addresses to allow them to connect to the ApsaraDB RDS instance.
- Configure an ECS security group: Add an ECS security group for the ApsaraDB RDS instance to allow ECS instances in the group to connect to the ApsaraDB RDS instance.

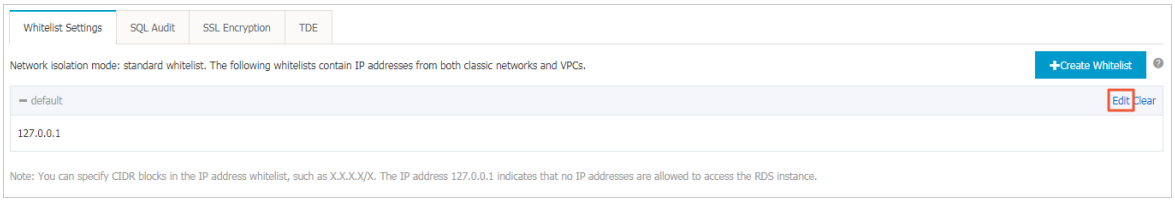
Precautions

- The default whitelist can be modified or cleared, but cannot be deleted.

- You can add up to 1,000 IP addresses or CIDR blocks to a whitelist. If you want to add a large number of IP addresses, we recommend that you merge them into CIDR blocks, such as 192.168.1.0/24.

Configure a standard IP address whitelist

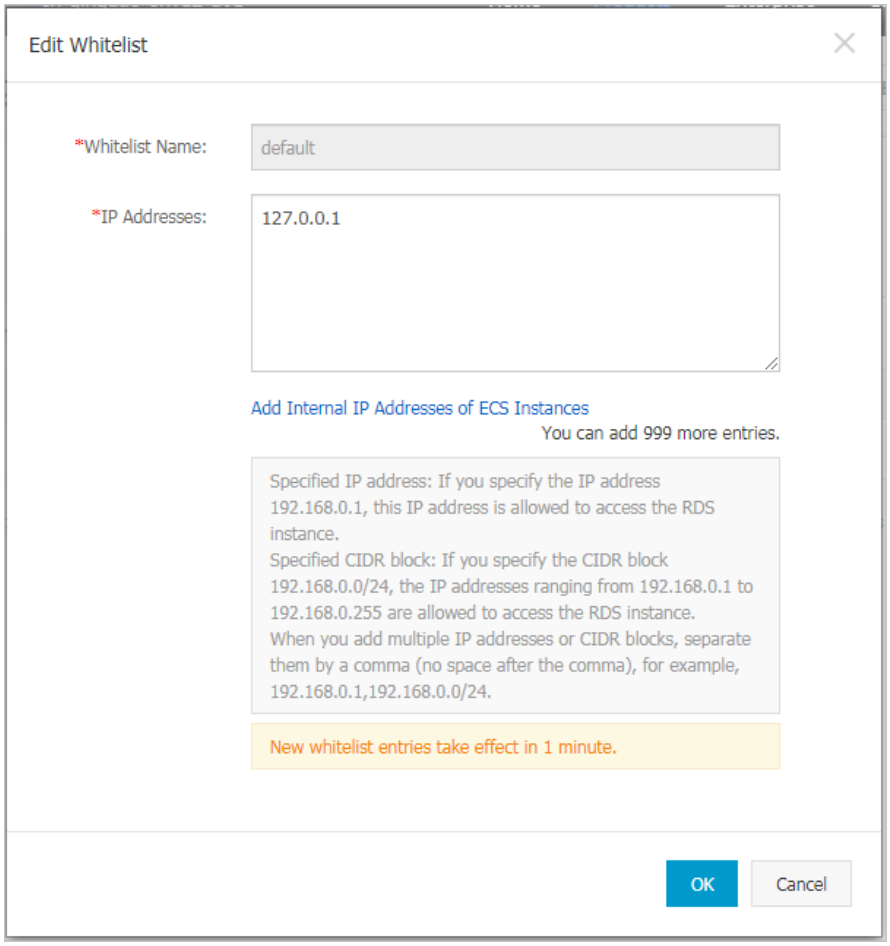
1. For more information, see Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find an instance. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
3. In the left-side navigation pane, click **Data Security**.
4. On the **Whitelist Settings** tab, click **Edit** corresponding to the **default** whitelist.



> **Note**
> - If you want to connect an ECS instance to an ApsaraDB RDS instance by using an internal endpoint, you must make sure that the two instances are in the same region and have the same network type. Otherwise, the connection fails.
> - You can also click **Create Whitelist** to create a new whitelist.

5. In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks that are allowed to access your ApsaraDB RDS instance, and then click **OK**.
   - If you add the CIDR block 10.10.10.0/24, all IP addresses in the 10.10.10.X format are allowed to access the ApsaraDB RDS instance.
   - If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
   - If you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all of the ECS instances that are created in your Alibaba Cloud account appear. Then, you can select the required IP addresses and add them to the whitelist.

> **Note** If you add a new IP address or CIDR block to the **default** whitelist, the default address 127.0.0.1 is deleted.



9.1.3.4.2. Create an account

After you create an ApsaraDB RDS instance and configure its whitelist, you must create a database and an account in the instance. This topic describes how to create privileged and standard accounts.

Context

ApsaraDB RDS for MySQL supports two types of database accounts: privileged and standard. You can manage all your accounts and databases in the ApsaraDB RDS console. For more information about permissions that can be granted to each type of account, see Account permissions.

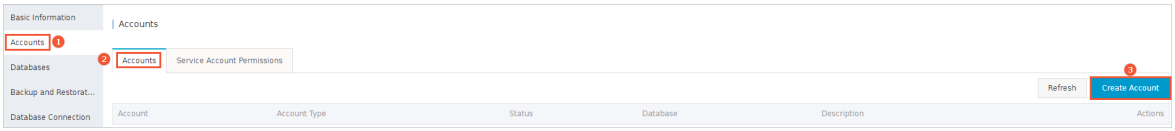
Account type	Description
Privileged account	• You can create and manage privileged accounts by using the ApsaraDB RDS console or API operations. • You can create only a single privileged account on each RDS instance. The privileged account can be used to manage all standard accounts and databases on the instance. • A privileged account allows you to manage permissions to a fine-grained level. For example, you can grant each standard account the permissions to query specific tables. • A privileged account has the permissions to disconnect all standard accounts on the instance.

Account type	Description
Standard account	• You can create and manage standard accounts by using the ApsaraDB RDS console, API operations, or SQL statements. • You can create up to 500 standard accounts for an instance. • You must manually grant standard accounts the specific database permissions. • You cannot use a standard account to create, manage, or disconnect other accounts from databases.

Account type	Number of databases	Number of tables	Number of accounts
Privileged account	Unlimited	< 200,000	Varies based on the kernel parameter settings of the instance.
Standard account	500	< 200,000	Varies based on the kernel parameter settings of the instance.

Create a privileged account

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Create Account**.



6. Configure the following parameters.

Parameter	Description
Database Account	Enter the account name. The account name must meet the following requirements: ◦ The name must be 1 to 16 characters in length. ◦ The name must start with a letter and end with a letter or digit. ◦ The name can contain lowercase letters, digits, and underscores (_).
Account Type	Select Privileged Account.
Password	Enter the password of the account. The password must meet the following requirements: ◦ The password must be 8 to 32 characters in length. ◦ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ◦ Special characters include ! @ # $ % ^& * ( ) _ + - =
Re-enter Password	Enter the password of the account again.

Parameter	Description
Description	Optional. Enter information about the account to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

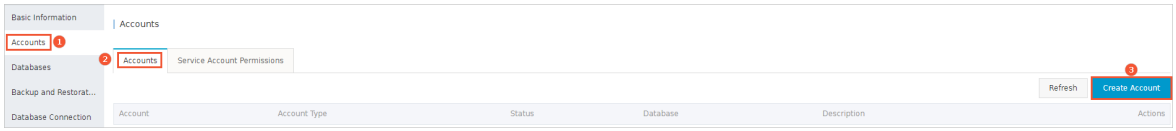
Reset the permissions of a privileged account

If an issue occurs on the privileged account, you can enter the password of the privileged account to reset permissions. For example, you can reset the permissions if the permissions are unexpectedly revoked.

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Find the privileged account, and click **Reset Permissions** in the **Actions** column.
6. Enter the password of the privileged account and click **OK**.

Create a standard account

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Create Account**.



6. Configure the following parameters.

Parameter	Description
Database Account	Enter the account name. The account name must meet the following requirements: ○ The name must be 1 to 16 characters in length. ○ The name must start with a letter and end with a letter or digit. ○ The name can contain lowercase letters, digits, and underscores (_).
Account Type	Select Standard Account.

Parameter	Description
Authorized Databases	Select one or more databases on which you want to grant permissions to the account. You can also leave this parameter empty at this time and authorize databases after the account is created. i. Select one or more databases from the Unauthorized Databases section and click **Add** to add them to the Authorized Databases section. ii. In the Authorized Databases section, select the **Read/Write**, **Read-only**, **DDL Only**, or **DML Only** permissions on each authorized database. If you want to grant the same permissions on multiple databases to the account, click the button in the upper-right corner of the section. The button may appear as **Set All to Read/Write**.
Password	Enter the password of the account. The password must meet the following requirements: ○ The password must be 8 to 32 characters in length. ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ Special characters include ! @ # $ % ^ & * ( ) _ + - =
Re-enter Password	Enter the password of the account again.
Description	Optional. Enter information about the account to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

Account permissions

Account type	Authorization type	Permission				
Privileged account	-	SELECT	INSERT	UPDATE	DELETE	CREATE
		DROP	RELOAD	PROCESS	REFERENCES	INDEX
		ALTER	CREATE TEMPORARY TABLES	LOCK TABLES	EXECUTE	REPLICATION SLAVE
		REPLICATION CLIENT	CREATE VIEW	SHOW VIEW	CREATE ROUTINE	ALTER ROUTINE
		CREATE USER	EVENT	TRIGGER	-	-

Account type	Authorization type	Permission				
Standard account	Read-only	SELECT	LOCK TABLES	SHOW VIEW	PROCESS	REPLICATION SLAVE
		REPLICATION CLIENT	-	-	-	-
	Read/write	SELECT	INSERT	UPDATE	DELETE	CREATE
		DROP	REFERENCES	INDEX	ALTER	CREATE TEMPORARY TABLES
		LOCK TABLES	EXECUTE	CREATE VIEW	SHOW VIEW	CREATE ROUTINE
		ALTER ROUTINE	EVENT	TRIGGER	PROCESS	REPLICATION SLAVE
		REPLICATION CLIENT	-	-	-	-
	DDL-only	CREATE	DROP	INDEX	ALTER	CREATE TEMPORARY TABLES
		LOCK TABLES	CREATE VIEW	SHOW VIEW	CREATE ROUTINE	ALTER ROUTINE
		PROCESS	REPLICATION SLAVE	REPLICATION CLIENT	-	-
	DML-only	SELECT	INSERT	UPDATE	DELETE	CREATE TEMPORARY TABLES
		LOCK TABLES	EXECUTE	SHOW VIEW	EVENT	TRIGGER
		PROCESS	REPLICATION SLAVE	REPLICATION CLIENT	-	-

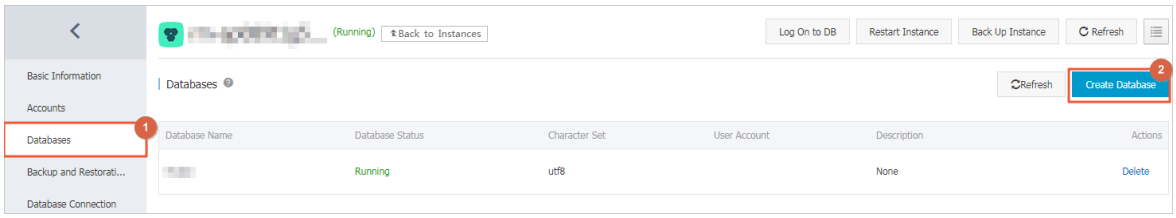
9.1.3.4.3. Create a database

After you create an ApsaraDB RDS instance and configure its whitelist, you must create a database and an account in the instance.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Databases**.

5. Click **Create Database**.



6. Configure the following parameters.

Parameter	Description
Database Name	○ The name must be 1 to 64 characters in length. ○ The name must start with a letter and end with a letter or digit. ○ The name can contain lowercase letters, digits, underscores (_), and hyphens (-). ○ The name must be unique within the instance.
Supported Character Sets	Select utf8, gbk, latin1, utf8mb4, or all. If you want to use other character sets, select **all**, and then select the required character set from the list.
Description	Optional. Enter information about the database to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

9.1.3.5. Connect to an ApsaraDB RDS for MySQL instance

After you complete the initial configuration of your ApsaraDB RDS for MySQL instance, you can connect to it from an Elastic Compute Service (ECS) instance or an on-premises client.

Context

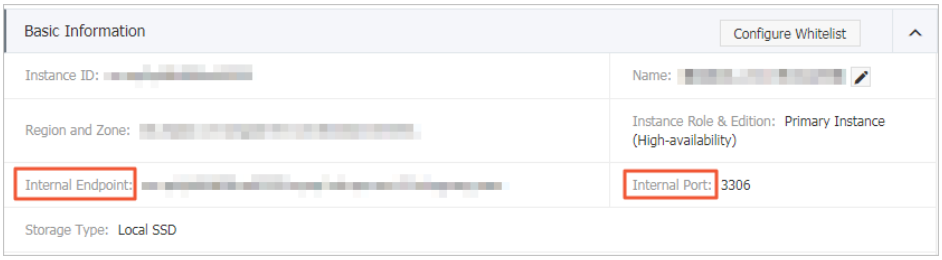
After you perform operations such as Create an instance, Configure a whitelist, and Create an account, you can use a general database client or configure the endpoint, port number, and account information in an application to connect to the MySQL instance.

If you need to connect an ECS instance to an ApsaraDB RDS instance, you must make sure that both instances are in classic networks or in the same VPC, and the IP address of the ECS instance is correctly configured in the ApsaraDB RDS whitelist.

Connect to an instance from a client

ApsaraDB RDS for MySQL is fully compatible with open source MySQL. You can connect to an ApsaraDB RDS instance from a database client by using a method similar to the method that you use to connect to an open source MySQL database. In the following example, the HeidiSQL client is used.

1. Start the HeidiSQL client.
2. In the lower-left corner of the Session manager dialog box, click **New**.
3. Enter information about the ApsaraDB RDS instance that you want to connect. The following table describes the required parameters.

Parameter	Description
Network type	Select the network type of the ApsaraDB RDS instance that you want to connect. For this example, select **MariaDB or MySQL (TCP/IP)**.
Hostname / IP	Enter the internal or public endpoint of the ApsaraDB RDS instance. ○ If your client is deployed on an ECS instance that is in the same region and has the same network type as the ApsaraDB RDS instance, use the internal endpoint. For example, if your ECS and ApsaraDB RDS instances are both in a VPC located in the China (Hangzhou) region, you can use the internal endpoint of the ApsaraDB RDS instance to create a secure connection. ○ In other scenarios, use the public endpoint. To view the internal and public endpoints and port numbers of the ApsaraDB RDS instance, perform the following operations: i. Log on to the ApsaraDB for RDS console. ii. Find the ApsaraDB RDS instance to which you want to connect and click its ID. iii. In the **Basic Information** section, view the internal endpoint and internal port number of the instance. 
User	Enter the username of the account that you use to connect to the ApsaraDB RDS instance.
Password	Enter the password of the account.
Port	If you connect to the instance over an internal network, enter the internal port number of the instance. If you connect to the instance over the Internet, enter the public port number of the instance.

4. Click **Open**. If the connection information is correct, you can connect to the instance.

9.1.4. Instances

9.1.4.1. Create an instance

This topic describes how to create one or more instances in the ApsaraDB RDS console.

Prerequisites

An Apsara Stack tenant account is obtained.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	Organization	The organization to which the instance belong.
	Resource Set	The resource set to which the instance belong.
Region	Region	The region in which you want to create the instance. Services in different regions cannot communicate over an internal network. After the instance is created, the region cannot be changed.
	Zone of Primary Node	The zone where the primary instance is deployed.
	Deployment Method	Specifies whether to deploy the primary and secondary instances in separate zones. ApsaraDB RDS supports **Multi-zone Deployment** and **Single-zone Deployment**. If you select **Multi-zone Deployment**, you must configure **Zone of Secondary Node**.
	Zone of Secondary Node	The zone where the secondary instance is deployed. This parameter is available only when **Deployment Method** is set to **Multi-zone Deployment**. **Note** If you select the same zone for both the primary and secondary instances, the deployment is equivalent to single-zone deployment.
Specifications	Quantity	The number of ApsaraDB RDS instances that you want to create. Default value: 1.
	Instance Name	The name of the instance. ○ The name must be 2 to 64 characters in length. ○ The name must start with a letter. ○ The name can contain letters, digits, and the following special characters: _ - : ○ The name cannot start with http:// or https://.
	Network Type	The connection type of the instance. ApsaraDB RDS instances support the following connection types: ○ **Internet Connection**: ApsaraDB RDS instances of this connection type can be connected over the Internet. ○ **Internal Network**: ApsaraDB RDS instances of this connection type can be connected over an internal network. **Note** The value of this parameter cannot be changed after the instance is created. Proceed with caution.
	Database Engine	The database engine of the instance. Set the value to **MySQL**.
	Engine Version	The version of the database engine. Valid values: ○ 5.6 ○ 5.7

Section	Parameter	Description
	Edition	The edition of the instance. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Type	The storage type of the instance. Local and standard SSDs are supported. **Note** Standard SSDs are supported only for MySQL 5.7 instances on High-availability Edition.
	Encrypted	Specifies whether to encrypt the standard SSD. This parameter is available only when **Storage Type** is set to **Standard SSD**. If you select Encrypted, you must specify the **Key** parameter. If you do not have a key, you must first create one in Key Management Service (KMS). For more information, see Create a CMK in *KMS User Guide*.
	Key	The key used to encrypt the standard SSD. This parameter is available only when you select **Encrypted**.
	Instance Type	The instance type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Capacity	The storage capacity of the instance, which includes the space to store data, system files, binlog files, and transaction files. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
Network	**Network Type**	The network type of the instance. ApsaraDB RDS instances support the following network types: ○ **Classic Network**: Cloud services in the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. **Note** Instances that use standard SSDs cannot be deployed in the classic network. ○ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Alibaba Cloud. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for enhanced security.
	VPC	The VPC in which you want to create the instance. **Note** You must specify this parameter when **Network Type** is set to **VPC**.
	vSwitch	The vSwitch in the VPC. **Note** You must specify this parameter when **Network Type** is set to **VPC**.
	IP Address Whitelist	The IP addresses that are allowed to connect to the ApsaraDB RDS instance.

4. After you configure the preceding parameters, click **Submit**.

9.1.4.2. Create an ApsaraDB RDS for MySQL instance with standard or enhanced SSDs

Cloud disks are block-level data storage products provided by Alibaba Cloud for ECS. They provide low latency, high performance, high durability, and high reliability. This topic describes how to create an instance with standard or enhanced SSDs in the ApsaraDB RDS console.

Prerequisites

The instance runs MySQL 5.7 on RDS High-availability Edition.

Context

An ApsaraDB RDS instance with standard or enhanced SSDs uses a distributed triplicate mechanism to ensure high data reliability. If service disruptions occur within a zone due to reasons such as a hardware failure, data in that zone is copied to an unaffected disk in another zone to ensure data availability.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	Region	The region in which you want to create the instance. Services in different regions cannot communicate over an internal network. After the instance is created, the region cannot be changed.
	Zone of Primary Node	The zone where the primary instance is deployed.
	Deployment Method	Specifies whether to deploy the primary and secondary instances in separate zones. ApsaraDB RDS supports **Multi-zone Deployment** and **Single-zone Deployment**. If you select **Multi-zone Deployment**, you must configure **Zone of Secondary Node**.
	Zone of Secondary Node	The zone where the secondary instance is deployed. This parameter is available only when **Deployment Method** is set to **Multi-zone Deployment**. **Note** If you select the same zone for both the primary and secondary instances, the deployment is equivalent to single-zone deployment.
	Quantity	The number of ApsaraDB RDS instances that you want to create. Default value: 1.

Section	Parameter	Description
Specifications	**Instance Name**	The name of the instance. ○ The name must be 2 to 64 characters in length ○ The name must start with a letter. ○ The name can contain letters, digits, and the following special characters: _ - : ○ The name cannot start with http:// or https://.
	Network Type	The connection type of the instance. ApsaraDB RDS instances support the following connection types: ○ **Internet Connection**: ApsaraDB RDS instances of this connection type can be connected over the Internet. ○ **Internal Network**: ApsaraDB RDS instances of this connection type can be connected over an internal network. **Note** The value of this parameter cannot be changed after the instance is created. Proceed with caution.
	Database Engine	The database engine of the instance. Select **MySQL**.
	Engine Version	The version of the database engine. Select **5.7**.
	Edition	The edition of the instance. Select **High-availability**.
	Storage Type	The storage type of the instance. Set the value to **Standard SSD**.
	Encrypted	Specifies whether to encrypt the standard SSD. This parameter is available only when **Storage Type** is set to **Standard SSD**. If you select Encrypted, you must specify the **Key** parameter. If you do not have a key, you must first create one in Key Management Service (KMS). For more information, see Create a CMK in *KMS User Guide*.
	Key	The key used to encrypt the standard SSD. This parameter is available only when you select **Encrypted**.
	Instance Type	The instance type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Capacity	The storage capacity of the instance, which includes the space to store data, system files, binlog files, and transaction files. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.

Section	Parameter	Description
Network	Network Type	The network type of the instance. ApsaraDB RDS instances support the following network types: ○ **Classic Network**: Cloud services in the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ○ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Alibaba Cloud. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for improved security.
	VPC	The VPC in which you want to create the instance. **Note** You must specify this parameter when **Network Type** is set to **VPC**.
	vSwitch	The vSwitch in the VPC. **Note** You must specify this parameter when **Network Type** is set to **VPC**.
	IP Address Whitelist	The IP addresses that are allowed to connect to the instance.

4. Click **Submit**.

9.1.4.3. View basic information of an instance

This topic describes how to view the details of an ApsaraDB RDS instance, such as its basic information, internal network connection information, status, and configurations.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. Use one of the following methods to go to the **Basic Information** page of an instance:
   - On the **Instances** page, click the ID of an instance to go to the **Basic Information** page.
   - On the **Instances** page, click **Manage** in the **Actions** column corresponding to an instance to go to the **Basic Information** page.

9.1.4.4. Restart an instance

This topic describes how to manually restart an ApsaraDB RDS for MySQL instance. This applies if the number of connections exceeds a specific threshold or if an instance has performance issues.

Prerequisites

The instance is in the **Running** state.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.

3. Click **Restart Instance** in the upper-right corner.

> **Note** When you restart an instance, applications are disconnected from the instance. We recommend that you make appropriate service arrangements before you restart an instance. Proceed with caution.

4. In the Restart Instance message, click **Confirm**.

9.1.4.5. Change the specifications of an instance

This topic describes how to change specifications of your instance, such as the instance type and storage space, if the specifications do not meet the requirements of your application.

Procedure

1. For more information, see Log on to the ApsaraDB RDS console.
2. Find your ApsaraDB RDS instance and click its ID.
3. In the **Configure Information** section of the **Basic Information** page, click **Change Specifications**.
4. On the **Change Specifications** page, set **Edition**, **Instance Type**, and **Storage Capacity**.
5. Click **Submit**.

9.1.4.6. Set a maintenance window

This topic describes how to set a maintenance window for an ApsaraDB RDS instance.

Context

To ensure the stability of ApsaraDB RDS instances, the backend system performs maintenance of the instances at irregular intervals. The default maintenance window is from 02:00 to 06:00. You can set the maintenance window to the off-peak period of your business to avoid impact on business.

Precautions

- To ensure the stability of the maintenance process, the instance changes to the **Maintaining Instance** state before the maintenance window. When the instance is in this state, access to data in the database and query operations such as performance monitoring are not affected. However, apart from account and database management and IP address whitelist configuration, modification operations such as upgrade, downgrade, and restart are temporarily unavailable.
- During the maintenance window, one or two transient connections may occur. Make sure that you configure automatic reconnection policies for your applications to avoid service disruptions.

Procedure

1. Connect to your ApsaraDB RDS instance. For more information, see Log on to the ApsaraDB RDS console.
2. Click the ID of an instance or click **Manage** in the **Actions** column corresponding to the instance.
3. In the **Configuration Information** section, click **Configure** to the right of **Maintenance Window**.
4. Select a maintenance window and click **Save**.

> **Note** The maintenance window is in UTC+8.

9.1.4.7. Change the data replication mode

You can set the data replication mode between primary and secondary ApsaraDB RDS instances to improve database availability.

Prerequisites

The ApsaraDB RDS instance uses local SSDs.

Context

- Semi-sync

  After an application-initiated update is complete on the primary instance of a cluster, logs are synchronized to all secondary instances. This transaction is considered committed after at least one secondary instance has received the logs, regardless of whether the secondary instance finishes executing the updates specified in the logs.

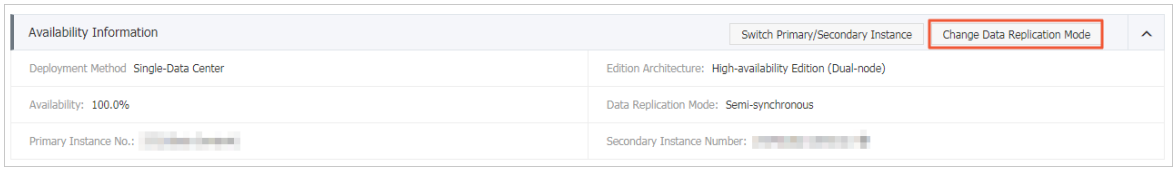
  If the secondary instances are unavailable or a network exception occurs between the primary and secondary instances, semi-synchronous replication degrades to the Asynchronous mode.

- Asynchronous

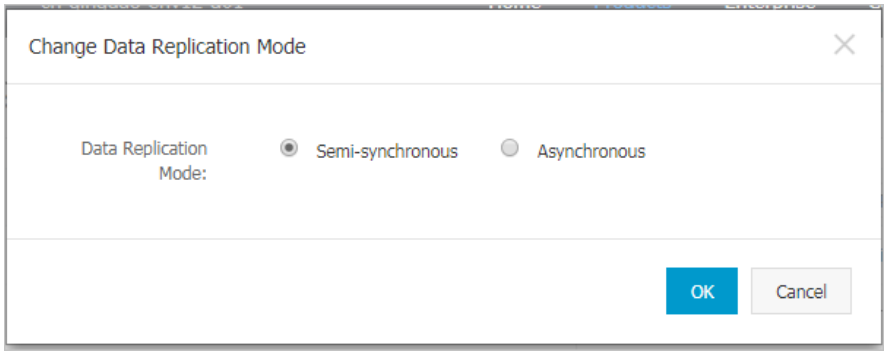
  When your application initiates a request to add, delete, or modify data, the primary instance responds to your application immediately after it completes the operation. At the same time, the primary instance starts to asynchronously replicate data to its secondary instances. During asynchronous data replication, operations on the primary instance are not affected if the secondary instances are unavailable. If the primary instance is unavailable, data remains consistent.

Procedure

1. For more information, see Log on to the ApsaraDB RDS console.
2. Find your ApsaraDB RDS instance and click its ID.
3. In the left-side navigation pane, click **Service Availability**.
4. Click **Change Data Replication Mode**.



5. In the Change Data Replication Mode dialog box, select a data replication mode and click **OK**.



9.1.4.8. Release an instance

This topic describes how to manually release an instance.

Precautions

- Only instances in the running state can be released.
- After an instance is released, the instance data is immediately deleted. We recommend that you back up instance data before you release an instance.

Procedure

1. Log on to the ApsaraDB RDS console.
2. In the Actions column corresponding to the instance you want to release, choose **More > Release Instance**.
3. In the **Release Instance** message, click **Confirm**.

9.1.4.9. Upgrade the minor version of an instance

ApsaraDB RDS for MySQL supports automatic and manual updates of the minor version. These updates increase performance, unveil new features, and fix known issues.

Overview

ApsaraDB RDS for MySQL automatically upgrades the minor version by default. You can log on to the ApsaraDB RDS console, go to the **Basic Information** page of your ApsaraDB RDS instance, and then view the current **Minor Version Upgrade Mode** in the Configuration Information section.

- **Automatic Upgrade**: When a new minor version is released, the system automatically upgrades the minor version of your instance during the specified maintenance window. For more information, see Set a maintenance window.
- **Manual Upgrade**: You can manually upgrade the minor version on the **Basic Information** page. For more information, see Manually upgrade the minor version.

Precautions

- When you upgrade the minor version of your ApsaraDB RDS instance, a 30-second network interruption may occur. We recommend that you upgrade the minor version during off-peak hours or make sure that your applications are configured with automatic reconnection policies.
- The minor version of your ApsaraDB RDS instance cannot be downgraded after it is upgraded.
- After you upgrade the specifications of your ApsaraDB RDS instance, the minor version of the instance is upgraded.

Configure the minor version upgrade mode

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section of the Basic Information page, click **Configure** to the right of **Minor Version Upgrade Mode**.
5. In the Set Minor Version Upgrade Mode dialog box,select **Auto** or **Manual**, and click **OK**.

Manually upgrade the minor version

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section of the page, click **Upgrade Minor Version**.

   > **Note** The **Upgrade Minor Version** button is displayed only when a new minor version is available.

5. In the dialog box that appears, specify the upgrade time and click **OK**.

FAQ

- Q: After I upgraded the minor version of my ApsaraDB RDS instance, why does the SELECT @@version statement still return the source minor version that I used before the upgrade?

  A: The SELECT @@version statement returns the minor version of Alibaba Cloud, not the minor version of the ApsaraDB RDS for MySQL instance. You need to execute the `show variables like '%rds_release_date%'` statement to view the minor version of your instance.

- Q: When an upgrade takes effect, is my instance upgraded only to the next minor version?

  A: No, when an upgrade takes effect, your instance is upgraded to the latest minor version.

9.1.4.10. Modify parameters of an instance

This topic describes how to view and modify the values of some parameters and query parameter modification records in the console.

Precautions

- To ensure instance stability, you can select specific parameters to modify in the ApsaraDB RDS console.
- When you modify parameters on the **Editable Parameters** tab, refer to the **Value Range** column corresponding to each parameter.
- After some parameters are modified, you must restart your ApsaraDB RDS instance for the changes to take effect. For more information, see the **Force Restart** column on the **Editable Parameters** tab. We recommend that you modify the parameters of an instance during off-peak hours and make sure that your applications are configured with automatic reconnection policies.

Modify parameters

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Parameters**.
5. You can perform the following operations:

   Export the parameter settings of the ApsaraDB RDS instance to your computer.

   On the Editable Parameters tab, click **Export Parameters**. The parameter settings of the ApsaraDB RDS instance are exported as a TXT file to your computer.

   Modify and import the parameter settings.

   i. After you have modified parameters in the exported parameter file, click **Import Parameters** and copy the parameter settings to the field.

   ii. Click **OK**.

   iii. In the upper-right corner of the page, click **Apply Changes**.

   > **Note**
   > - If the new parameter value takes effect only after an instance restarts, the system prompts you to restart the ApsaraDB RDS instance. We recommend that you restart the ApsaraDB RDS instance during off-peak hours and make sure that your applications are configured with automatic reconnection policies.
   > - Before the new parameter values are applied, you can click **Cancel Changes** to cancel them.

   Modify a single parameter.

i. On the **Editable Parameters** tab, find the parameter that you want to reconfigure, and click the icon in the **Actual Value** column.
ii. Enter a new value based on the prompted value range.
iii. Click **Confirm**.
iv. In the upper-right corner of the page, click **Apply Changes**.

> Note
> - If the new parameter value takes effect only after an instance restarts, the system prompts you to restart the ApsaraDB RDS instance. We recommend that you restart the ApsaraDB RDS instance during off-peak hours and make sure that your applications are configured with automatic reconnection policies.
> - Before the new parameter value is applied, you can click **Cancel Changes** to cancel it.

View the parameter modification history

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Parameters**.
5. On the Parameters page, click the **Edit History** tab.
6. Select a time range and then click **Search**.

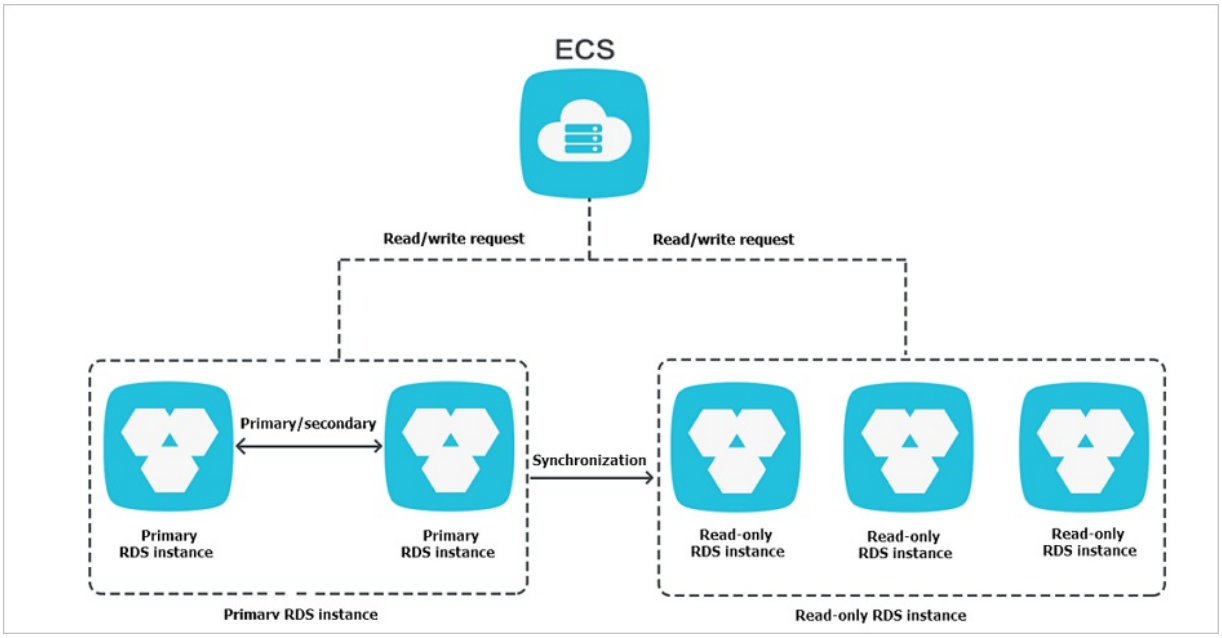
9.1.4.11. Read-only instances

9.1.4.11.1. Overview of read-only instances

ApsaraDB RDS for MySQL allows you to create read-only instances. In scenarios where an instance has a small number of write requests but a large number of read requests, you can create read-only instances to distribute database access loads away from the primary instance. This topic describes the features and limits of read-only instances.

To scale up the reading capability and distribute database access loads, you can create one or more read-only instances in a region. Read-only instances can increase the application throughput when a large amount of data is being read.

A read-only instance with a single physical node and no backup node uses the native replication capability of MySQL to synchronize changes from the primary instance to all its read-only instances. Read-only instances must be in the same region as the primary instance but do not have to be in the same zone as the primary instance. The following figure shows the topology of read-only instances.



Read-only instances have the following features:

- Specifications of a read-only instance can be different from those of the primary instance and can be changed at any time. This facilitates elastic scaling.
- Read-only instances do not require account or database maintenance. Account and database information is synchronized from the primary instance.
- The whitelists of read-only instances can be independently configured.
- System performance monitoring is provided.

  ApsaraDB RDS provides up to 20 system performance monitoring views, including those for disk capacity, IOPS, connections, CPU utilization, and network traffic. You can view the load of instances.
- ApsaraDB RDS provides a variety of optimization recommendations, such as storage engine check, primary key check, large table check, and check for excessive indexes and missing indexes. You can optimize your databases based on the optimization recommendations and specific applications.

9.1.4.11.2. Create a read-only instance

You can create read-only instances of different specifications based on your business requirements.

Precautions

- A maximum of five read-only instances can be created for a primary instance.
- Backup settings and temporary backup are not supported.
- Instance restoration is not supported.
- Data migration to read-only instances is not supported.
- Database creation and deletion are not supported.
- Account creation, deletion, authorization, and password changes are not supported.
- After a read-only instance is created, you cannot restore data by directly overwriting the primary instance with a backup set.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic**

Information page.

4. In the **Distributed by Instance Role** section on the right side of the **Basic Information** page, click **Create Read-only Instance**.
5. On the **Create Read-only Instance** page, configure the following parameters.

Section	Parameter	Description
Region	**Region**	The region in which you want to create the read-only instance.
Specifications	**Database Engine**	The database engine of the read-only instance, which is the same as that of the primary instance and cannot be changed.
	Engine Version	The version of the database engine, which is the same as that of the primary instance and cannot be changed.
	Edition	Set the value to **Read-only**.
	Instance Type	The instance type of the read-only instance. The instance type of the read-only instance can be different from that of the primary instance, and can be changed at any time to facilitate flexible upgrade and downgrade.
	Storage Capacity	The storage capacity of the read-only instance. To ensure sufficient I/O throughput for data synchronization, we recommend that you select at least the same instance type and storage capacity as the primary instance for the read-only instance. Valid values: 20 to 6000. Unit: GB. The value is in 1 GB increments.
Network Type	**Network Type**	The network type of the read-only instance, which is the same as that of the primary instance and cannot be changed.
	VPC	The VPC in which you want to create the read-only instance.
	vSwitch	The vSwitch in the VPC.

6. After you configure the preceding parameters, click **Submit**.

9.1.4.11.3. View details of read-only instances

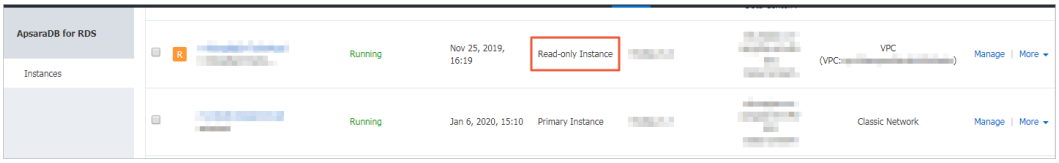
This topic describes how to view details of read-only instances. You can go to the Basic Information page of a read-only instance from the Instances page or from the read-only instance list of the primary instance. Read-only instances are managed in the same way as primary instances. The read-only instance management page shows the management operations that can be performed.

View details of a read-only instance from the Instances page

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, click the ID of a read-only instance. The **Basic Information** page appears.

   In the instance list, Instance Role of read-only instances is displayed as Read-only Instance, as shown in View a read-only instance.

   View a read-only instance



View details of a read-only instance from the Basic Information page of the primary instance

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. On the **Basic Information** page, move the pointer over the number below **Read-only Instance** in the **Distributed by Instance Role** section. The ID of the read-only instance is displayed.
5. Click the ID of the read-only instance to go to the read-only instance management page.

9.1.5. Accounts

9.1.5.1. Create an account

After you create an ApsaraDB RDS instance and configure its whitelist, you must create a database and an account in the instance. This topic describes how to create privileged and standard accounts.

Context

ApsaraDB RDS for MySQL supports two types of database accounts: privileged and standard. You can manage all your accounts and databases in the ApsaraDB RDS console. For more information about permissions that can be granted to each type of account, see Account permissions.

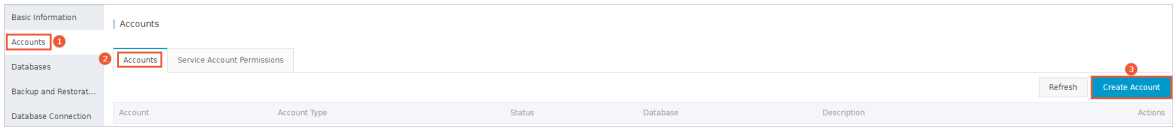
Account type	Description
Privileged account	• You can create and manage privileged accounts by using the ApsaraDB RDS console or API operations. • You can create only a single privileged account on each RDS instance. The privileged account can be used to manage all standard accounts and databases on the instance. • A privileged account allows you to manage permissions to a fine-grained level. For example, you can grant each standard account the permissions to query specific tables. • A privileged account has the permissions to disconnect all standard accounts on the instance.
Standard account	• You can create and manage standard accounts by using the ApsaraDB RDS console, API operations, or SQL statements. • You can create up to 500 standard accounts for an instance. • You must manually grant standard accounts the specific database permissions. • You cannot use a standard account to create, manage, or disconnect other accounts from databases.

Account type	Number of databases	Number of tables	Number of accounts
Privileged account	Unlimited	< 200,000	Varies based on the kernel parameter settings of the instance.

Account type	Number of databases	Number of tables	Number of accounts
Standard account	500	< 200,000	Varies based on the kernel parameter settings of the instance.

Create a privileged account

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Create Account**.



6. Configure the following parameters.

Parameter	Description
Database Account	Enter the account name. The account name must meet the following requirements: ○ The name must be 1 to 16 characters in length. ○ The name must start with a letter and end with a letter or digit. ○ The name can contain lowercase letters, digits, and underscores (_).
Account Type	Select Privileged Account.
Password	Enter the password of the account. The password must meet the following requirements: ○ The password must be 8 to 32 characters in length. ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ Special characters include ! @ # $ % ^& * ( ) _ + - =
Re-enter Password	Enter the password of the account again.
Description	Optional. Enter information about the account to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

Reset the permissions of a privileged account

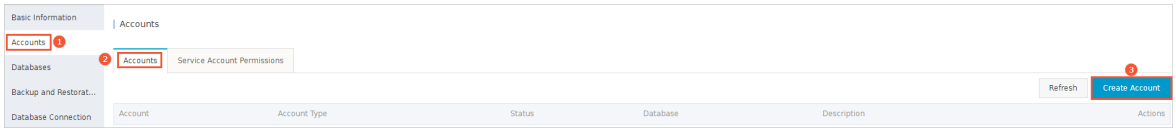
If an issue occurs on the privileged account, you can enter the password of the privileged account to reset permissions. For example, you can reset the permissions if the permissions are unexpectedly revoked.

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.

4. In the left-side navigation pane, click **Accounts**.
5. Find the privileged account, and click **Reset Permissions** in the **Actions** column.
6. Enter the password of the privileged account and click **OK**.

Create a standard account

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Create Account**.



6. Configure the following parameters.

Parameter	Description
Database Account	Enter the account name. The account name must meet the following requirements: ○ The name must be 1 to 16 characters in length. ○ The name must start with a letter and end with a letter or digit. ○ The name can contain lowercase letters, digits, and underscores (_).
Account Type	Select Standard Account.
Authorized Databases	Select one or more databases on which you want to grant permissions to the account. You can also leave this parameter empty at this time and authorize databases after the account is created. i. Select one or more databases from the Unauthorized Databases section and click **Add** to add them to the Authorized Databases section. ii. In the Authorized Databases section, select the **Read/Write**, **Read-only**, **DDL Only**, or **DML Only** permissions on each authorized database. If you want to grant the same permissions on multiple databases to the account, click the button in the upper-right corner of the section. The button may appear as **Set All to Read/Write**.
Password	Enter the password of the account. The password must meet the following requirements: ○ The password must be 8 to 32 characters in length. ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ Special characters include ! @ # $ % ^ & * ( ) _ + - =
Re-enter Password	Enter the password of the account again.
Description	Optional. Enter information about the account to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

Account permissions

Account type	Authorization type	Permission				
Privileged account	-	SELECT	INSERT	UPDATE	DELETE	CREATE
		DROP	RELOAD	PROCESS	REFERENCES	INDEX
		ALTER	CREATE TEMPORARY TABLES	LOCK TABLES	EXECUTE	REPLICATION SLAVE
		REPLICATION CLIENT	CREATE VIEW	SHOW VIEW	CREATE ROUTINE	ALTER ROUTINE
		CREATE USER	EVENT	TRIGGER	-	-
Standard account	**Read-only**	SELECT	LOCK TABLES	SHOW VIEW	PROCESS	REPLICATION SLAVE
		REPLICATION CLIENT	-	-	-	-
	Read/write	SELECT	INSERT	UPDATE	DELETE	CREATE
		DROP	REFERENCES	INDEX	ALTER	CREATE TEMPORARY TABLES
		LOCK TABLES	EXECUTE	CREATE VIEW	SHOW VIEW	CREATE ROUTINE
		ALTER ROUTINE	EVENT	TRIGGER	PROCESS	REPLICATION SLAVE
		REPLICATION CLIENT	-	-	-	-
	DDL-only	CREATE	DROP	INDEX	ALTER	CREATE TEMPORARY TABLES
		LOCK TABLES	CREATE VIEW	SHOW VIEW	CREATE ROUTINE	ALTER ROUTINE
		PROCESS	REPLICATION SLAVE	REPLICATION CLIENT	-	-
	DML-only	SELECT	INSERT	UPDATE	DELETE	CREATE TEMPORARY TABLES
		LOCK TABLES	EXECUTE	SHOW VIEW	EVENT	TRIGGER
		PROCESS	REPLICATION SLAVE	REPLICATION CLIENT	-	-

9.1.5.2. Reset the password

You can use the ApsaraDB RDS console to reset the password of your database account.

Prerequisites

The instance is in the **Running** state.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. In the left-side navigation pane, click **Accounts**.
4. Find an account and click **Reset Password** in the Actions column.
5. In the dialog box that appears, enter and confirm the new password, and then click **OK**.

> **Note** The password must meet the following requirements:
>
> - The password must be 8 to 32 characters in length.
> - The password must contain at least three of the following characters: uppercase letters, lowercase letters, digits, and special characters.
> - Special characters include ! @ # $ % ^ & * ( ) _ + - =

9.1.5.3. Modify account permissions

You can modify the account permissions of your ApsaraDB RDS instance at any time.

Prerequisites

You can modify the permissions of a standard account. The permissions of privileged accounts can only be reset to the default settings and cannot be changed to a specific set of permissions.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Find an account and click **Edit Permissions** in the Actions column.
6. Configure the following parameters

Parameter	Description
Authorized Databases	In the **Unauthorized Databases** section, select a database and click **Add** to authorize the database. In the **Authorized Databases** section, select a database and click **Remove** to remove the permissions from the database.

Parameter	Description
Authorized Databases	You can set permissions on each database in the Authorized Database section. You can also click the button such as **Set All to Read/Write** in the upper-right corner to set the permissions of the account on all authorized databases. ○ **Read-only**: grants the database read-only permissions to the account. ○ **Read/Write**: grants the database read/write permissions to the account. ○ **DDL Only**: grants the database permissions of DDL operations to the account. ○ **DML Only**: grants the database permissions of DML operations to the account.

7. Click **OK**.

9.1.5.4. Delete an account

You can delete a database account in the ApsaraDB RDS console.

Prerequisites

You can use the console to delete privileged and standard accounts that are no longer used.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Find the account you want to delete and click **Delete** in the Actions column.
6. In the message that appears, click **Confirm**.

> **Note** Accounts in the **Processing** state cannot be deleted.

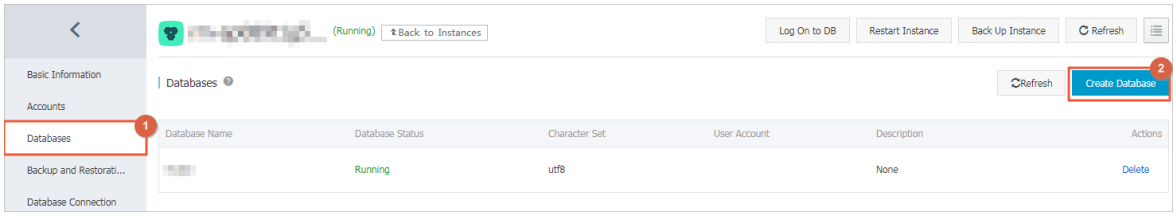
9.1.6. Databases

9.1.6.1. Create a database

After you create an ApsaraDB RDS instance and configure its whitelist, you must create a database and an account in the instance.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Databases**.
5. Click **Create Database**.



6. Configure the following parameters.

Parameter	Description
Database Name	○ The name must be 1 to 64 characters in length. ○ The name must start with a letter and end with a letter or digit. ○ The name can contain lowercase letters, digits, underscores (_), and hyphens (-). ○ The name must be unique within the instance.
Supported Character Sets	Select utf8, gbk, latin1, utf8mb4, or all. If you want to use other character sets, select **all**, and then select the required character set from the list.
Description	Optional. Enter information about the database to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

9.1.6.2. Delete a database

You can delete databases that are no longer used in the ApsaraDB RDS console.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Databases**.
5. Find the database you want to delete and click **Delete** in the Actions column.
6. In the Delete Database message, click **Confirm**.

9.1.7. Database connection

9.1.7.1. Change the endpoint and port number of an instance

This topic describes how to view and change the endpoint and port number of an instance.

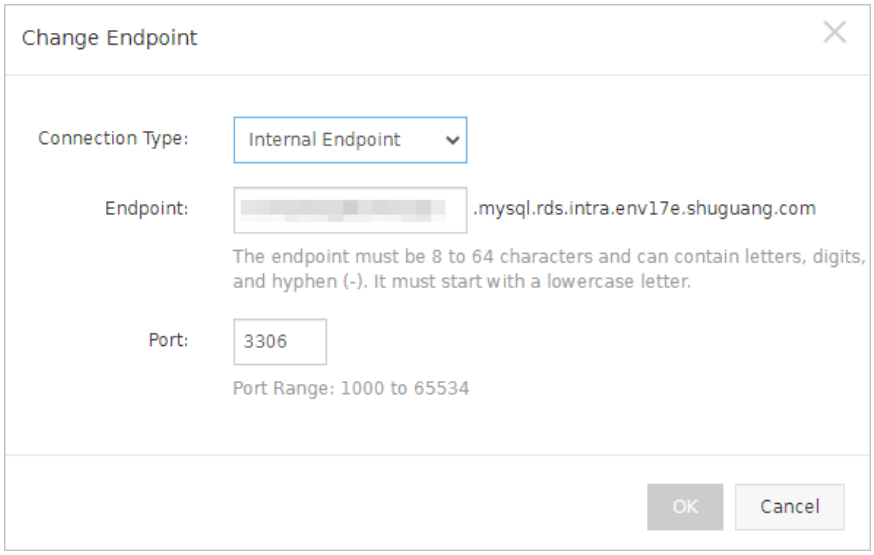
View the endpoint and port number

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.

You can view the internal endpoint and internal port of the instance in the **Database Connection** section.

Change the endpoint and port number

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the Database Connection section, click **Change Endpoint**.
6. In the dialog box that appears, set Connection Type, Endpoint, and Port, and click **OK**.



> **Note**
> - The prefix of the endpoint must be 8 to 64 characters in length and can contain only letters, digits, and hyphens (-). It must start with a lowercase letter.
> - The port number must be in the range of 1000 to 5999.

9.1.7.2. Log on to an ApsaraDB RDS instance by using DMS

This topic describes how to log on to an ApsaraDB RDS instance by using Data Management (DMS).

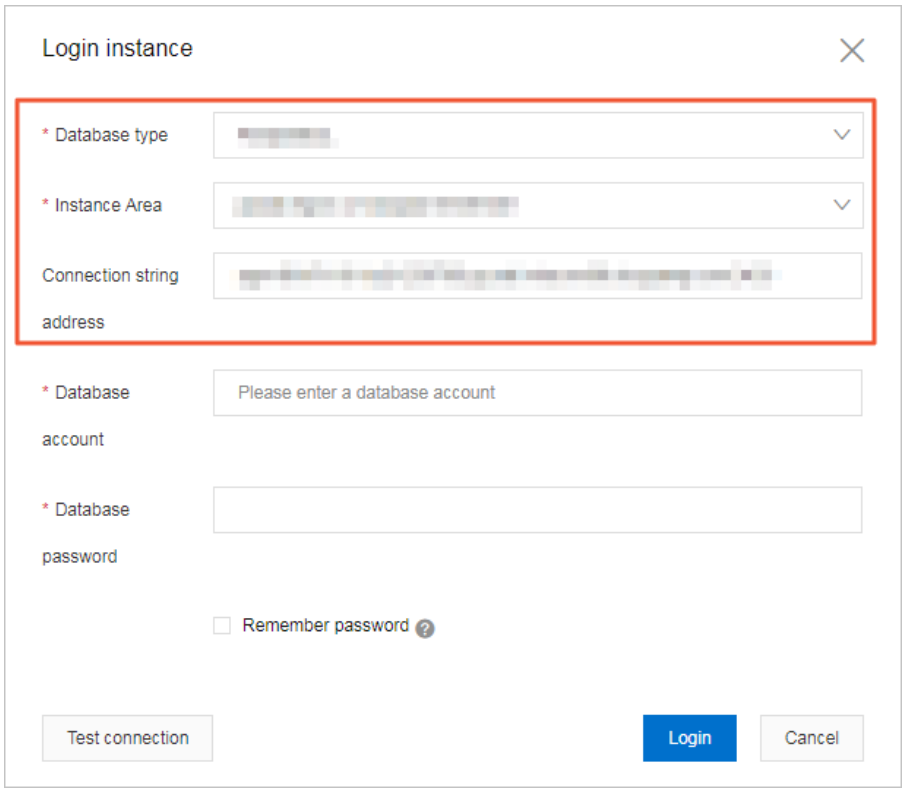
Prerequisites

The IP address whitelist is configured. For more information about how to configure an IP address whitelist, see Configure a whitelist.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Log On to DB** to go to the Database Logon page.
5. In the **Login instance** dialog box of the **DMS** console, check the **database type**, **instance area**, and **connection string address**. If the preceding information is correct, enter the **database account** and

database password, as shown in the following figure.



Parameter	Description
Database type	The database engine of the instance. By default, this parameter is set to the database engine of the instance that you want to access.
Instance Area	The region where the instance resides. By default, this parameter is set to the region where the current instance resides.
Connection string address	The endpoint and port number that are used to connect to the instance. By default, this parameter is set to the endpoint and port number of the current instance.
Database account	The account that is used to connect to the database.
Database Password	The password of the account that is used to connect to the database.

6. Click **Login**.

> **Note** If you want the browser to remember the password, select **Remember password** before you click **Login**.

9.1.7.3. Hybrid access from both the classic network and VPCs

This topic describes how to use the hybrid access solution of ApsaraDB RDS to change the network type of an instance from classic network to Virtual Private Network (VPC) without network interruptions.

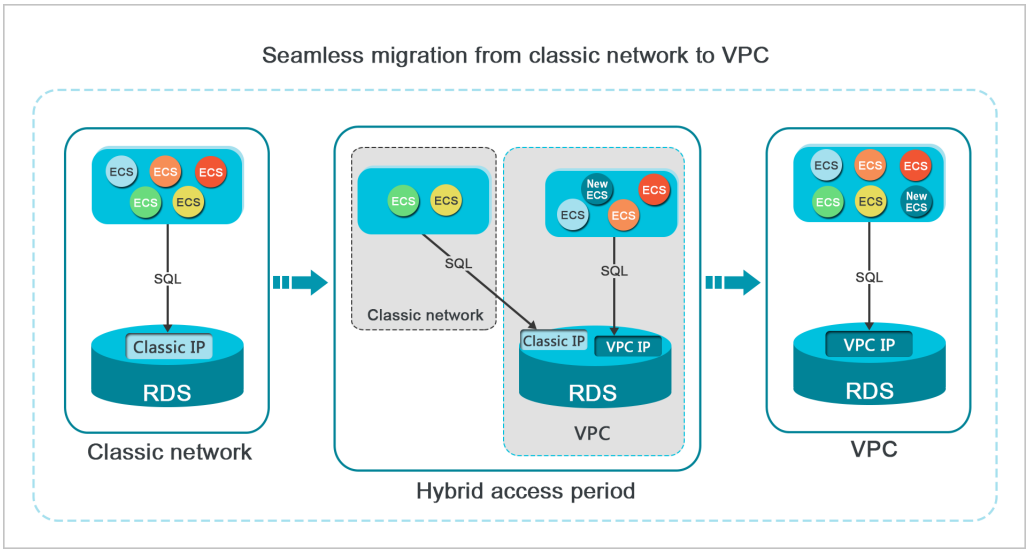
Background

In the past, when you change the network type of an ApsaraDB RDS instance from classic network to VPC, the internal endpoint of the instance would remain the same but the IP address bound to the endpoint would change to the corresponding IP address in the VPC. This change would cause a 30-second network interruption, and ECS instances within the classic network would not be able to access the ApsaraDB RDS instance by using the internal endpoint within this period. To smoothly change the network type, ApsaraDB RDS provides the hybrid access solution.

Hybrid access refers to the ability of an ApsaraDB RDS instance to be accessed by ECS instances in both the classic network and VPCs. During the hybrid access period, the ApsaraDB RDS instance reserves the original internal endpoint of the classic network and adds the internal endpoint of VPCs. This prevents network interruptions during the network type switchover.

For security and performance purposes, we recommend that you use only the internal VPC endpoint. Therefore, ApsaraDB RDS allows the configured hybrid access solution to remain valid only for a specific period. When the hybrid access period elapses, ApsaraDB RDS releases the internal classic network endpoint. In this case, your applications cannot connect to your ApsaraDB RDS instance by using the internal classic network endpoint. You must add the internal VPC endpoint to all your applications during the hybrid access period. This ensures a smooth network migration and avoids interruptions to your workloads.

For example, your company wants to use the hybrid access solution to change the network type from classic network to VPC. During the hybrid access period, some applications can access the database by using the internal endpoint of VPCs, and the other applications can access the database by using the original internal endpoint of the classic network. When all the applications access the database by using the internal endpoint of VPCs, the internal endpoint of the classic network can be released. The following figure illustrates the scenario.



Limits

During the hybrid access period, the instance has the following limits:

- The network type of the instance cannot be changed to classic network.
- The instance cannot be migrated to another zone.

Prerequisites

- The network type of the instance is classic network.
- Available VPCs and vSwitches exist in the zone where the instance resides.

Change the network type from classic network to VPC

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.

3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. On the Instance Connection tab, click **Switch to VPC**.
6. In the Switch to VPC dialog box, select a VPC and a vSwitch and specify whether to retain the endpoint used in the classic network.

   Determine whether to select **Reserve Original Classic Endpoint** based on the details described in the following table.

Operation	Description
Clear the Reserve Original Classic Network Endpoint option	The endpoint used in the classic network is replaced with an endpoint in the VPC. When you change the network type, a network interruption of about 30 seconds occurs, and the connection between ECS instances in the classic network and the ApsaraDB RDS instance are interrupted.
Select the Reserve Original Classic Network Endpoint option	The endpoint used in the classic network is retained, and a new endpoint to be used in the VPC is generated. In such cases, the ApsaraDB RDS instance runs in hybrid access mode. ECS instances in both the classic network and a VPC can connect to the ApsaraDB RDS instance over the internal network. When you change the network type, no network interruption occur. Connections between ECS instances in the classic network and the ApsaraDB RDS instance are available till the endpoint used in the classic network expires. Specify the expiration time of the classic network endpoint. You must add the new VPC endpoint to the ECS instance before the endpoint in the classic network expires. This ensures smooth network switchover.

7. Add the internal IP addresses of ECS instances in the selected VPC to a VPC whitelist. This allows the ECS instances to access the ApsaraDB RDS instance over the internal network. If no VPC whitelists are available, create a whitelist. For more information, see Configure a whitelist.

Change the expiration time for the original internal endpoint of the classic network

During the period in which your instance can be connected over the classic network or VPCs, you can specify the expiration time for the endpoint of the classic network. The setting takes effect immediately. For example, if the endpoint of the classic network expires on August 18, 2017 and you change the expiration time to 14 days later on August 15, 2017, the endpoint of the classic network is released on August 29, 2017.

To change the expiration time, perform the following operations:

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. Click **Change Expiration Time**.
6. In the **Change Expiration Time** dialog box, select an expiration time and click **OK**.

9.1.7.4. Change the network type of an instance

This topic describes how to change the network type of an ApsaraDB RDS instance between classic network and VPC.

Context

- Classic network: ApsaraDB RDS instances in the classic network are not isolated. Unauthorized access to these instances can be blocked only by whitelists.
- VPC: Each VPC is an isolated network. We recommend that you select the VPC network type because it is more secure than the classic network.

  You can configure route tables, CIDR blocks, and gateways in a VPC. To smoothly migrate applications to the cloud, you can use leased lines or VPNs to connect on-premises data center to a VPC to create a virtual data center.

Change the network type from VPC to classic network

Precautions

- After you change the network type from VPC to classic network, the internal endpoint of your ApsaraDB RDS instance remains unchanged. However, the IP address that is associated with the internal endpoint changes.
- After you change the network type from VPC to classic network, ECS instances in the same VPC as the ApsaraDB RDS instance can no longer connect to the ApsaraDB RDS instance by using the internal endpoint. You must update the endpoint for the applications deployed on the ECS instances.
- When you change the network type, a 30-second network interruption may occur. To avoid business interruption, change the network type during off-peak hours or make sure that your applications are configured with automatic reconnection policies.

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the Database Connection section, click **Switch to Classic Network**.
6. In the message that appears, click **OK**.

Change the network type from classic network to VPC

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the Database Connection section, click **Switch to VPC**.
6. In the Switch to VPC dialog box, select a VPC and vSwitch and specify whether to **Reserve Original Classic Network Endpoint**. Click OK. For more information about **Reserve Original Classic Network Endpoint**, see Hybrid access from both the classic network and VPCs.

9.1.7.5. Switch an ApsaraDB RDS for MySQL instance to a new VPC or vSwitch

This topic describes how to switch an ApsaraDB RDS for MySQL instance to a new VPC or vSwitch.

Prerequisites

The instance is deployed in a VPC.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the Database Connection section, click **Switch VSwitch**.
6. Select a VPC and a vSwitch, and then click **OK**.
7. In the message that appears, click **Switch**.

> **Note**
>
> - A 30-second network interruption occurs when you switch the VPC and vSwitch of an ApsaraDB RDS for MySQL instance. Make sure that your application is configured to automatically reconnect to the ApsaraDB RDS for MySQL instance.
> - We recommend that you clear the cache immediately after the instance is switched to a new VPC and vSwitch. Otherwise, data can only be read but cannot be written.

9.1.8. Database proxy

9.1.8.1. Dedicated proxy

This topic describes the dedicated proxy feature of ApsaraDB RDS for MySQL. This feature provides advanced features, such as read/write splitting, connection pooling, and transaction splitting.

Prerequisites

Your ApsaraDB RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 5.7 on RDS Enterprise Edition
- MySQL 5.7 on RDS High-availability Edition with local SSDs

Context

The dedicated proxy feature uses dedicated computing resources. This feature has the following benefits:

- A unified proxy endpoint is provided to connect to all the dedicated proxies that are enabled on your ApsaraDB RDS instance. This reduces maintenance costs by relieving the need to update the endpoints on your application. The proxy endpoint remains valid unless you release the dedicated proxies. For example, you may enable read/write splitting during peak hours, and then release read-only instances and disable read/write splitting after peak hours. In these cases, you do not need to update the endpoints on your application because the proxy endpoint remains connected.
- Dedicated proxies exclusively serve your ApsaraDB RDS instance and its read-only instances. You do not need to compete with other users for resources. This ensures service stability.
- Dedicated proxies are scalable. You can add dedicated proxies based on your business requirements to handle more workloads.

Limits

- Dedicated proxies do not support Secure Sockets Layer (SSL) encryption.
- Dedicated proxies do not support compression protocols.

Precautions

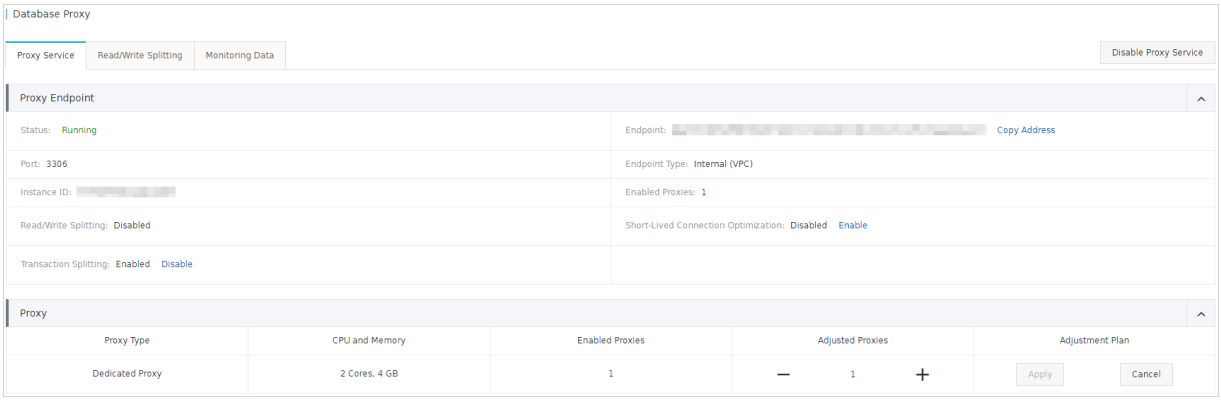
- When you change the specifications of your ApsaraDB RDS instance or its read-only instances, a network interruption may occur.
- If you connect your application to the proxy endpoint, all the requests that are encapsulated in transactions are routed to your ApsaraDB RDS instance. This applies if you do not enable the transaction splitting feature.
- If you use the proxy endpoint to implement read/write splitting, the read consistency of the requests that are not encapsulated in transactions cannot be guaranteed. If you require this read consistency, you must encapsulate these requests in transactions.
- If you connect your application to the proxy endpoint, the `SHOW PROCESSLIST` statement returns a combination of results from the primary ApsaraDB RDS instance and all of its read-only instances.
- If you execute multi-statements or stored procedures, the read/write splitting feature is disabled and all the subsequent requests over the current connection are routed to the primary ApsaraDB RDS instance. To enable the read/write splitting feature again, you must close the current connection and establish a new connection.
- The dedicated proxy feature supports the `/*FORCE_MASTER*/` and `/*FORCE_SLAVE*/` hints. However, requests that contain hints have the highest route priorities. Therefore, these requests are not constrained by consistency or transaction limits. Before you use these hints, you must check whether these hints are suitable for your workloads. In addition, these hints cannot contain statements such as `/*FORCE_SLAVE*/ set names utf8;`. Such statements can change environment variables. If you include such statements in these hints, errors may occur when you process your subsequent workloads.
- After you enable the dedicated proxy feature, each connection is replicated to the primary ApsaraDB RDS instance and all of its read-only instances in compliance with the 1:N connection model. We recommend that you specify the same connection specifications for these instances. If these instances have different connection specifications, the number of connections allowed varies based on the lowest connection specifications among these instances.
- If you create or restart a read-only instance after you enable the dedicated proxy feature, only the requests over a new connection are routed to the new or restarted read-only instance.
- The **max_prepared_stmt_count** parameter must be set to the same value for the primary ApsaraDB RDS instance and all of its read-only instances.

Enable the dedicated proxy feature

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. Click **Enable now**.

Overview of the Database Proxy page

After the dedicated proxy feature is enabled, you can use the generated proxy endpoint to implement features such as read/write splitting, connection pooling, and transaction splitting.



Section	Parameter	Description
Proxy Endpoint	**Instance ID**	The ID of the primary ApsaraDB RDS instance.
	Enabled Proxies	The number of enabled dedicated proxies. You can process more requests by enabling more dedicated proxies. After the public preview ends, you must pay for the proxies that you enabled.
	Read/Write Splitting	Specifies whether to enable the read/write splitting feature for the proxy endpoint. For more information, see Read/write splitting.
	Short-Lived Connection Optimization	The type of connection pool for the proxy endpoint. This feature is suitable for scenarios that PHP short-lived connections are established. For more information, see Short-lived connection optimization. **Note** You can click **Enable** or **Disable** to the right of Short-Lived Connection Optimization to enable or disable this feature.
	Transaction Splitting	Specifies whether to enable the transaction splitting feature for the proxy endpoint. For more information, see Transaction splitting. **Note** You can click **Enable** or **Disable** to the right of the Transaction Splitting parameter to enable or disable this feature.
	Endpoint	The proxy endpoint that is generated after the dedicated proxy feature is enabled. This endpoint connects to all the dedicated proxies that are enabled on the primary ApsaraDB RDS instance. The read/write splitting feature is also bound to this endpoint. **Note** You can click **Copy Address** to the right of Endpoint to copy the endpoint.
	Port	The port to connect to the proxy endpoint.
	Endpoint Type	The network type of the proxy endpoint.
	Proxy Type	The type of proxy that is enabled on the primary ApsaraDB RDS instance. Only the **Dedicated Proxy** type is supported.

Section	Parameter	Description
Proxy	CPU and Memory	The CPU and memory of the dedicated proxies. Only 2 Cores, 4 GB is supported.
	Enabled Proxies	The number of dedicated proxies that are enabled on the primary ApsaraDB RDS instance. A maximum of 60 dedicated proxies are supported. **Note** We recommend that you specify the number of dedicated proxies as the rounded-up integer of the total number of CPU cores of your ApsaraDB RDS instance and its read-only instances divided by 8. For example, if your ApsaraDB RDS instance has 8 CPU cores and its read-only instances have 4 CPU cores, the recommended number of dedicated proxies is 2 based on the following formula: (8 + 4)/8 = 1.5 (rounded up to 2).

Adjust the number of dedicated proxies

Note When you adjust the number of dedicated proxies, a network interruption occurs. Make sure that your applications are configured with automatic reconnection policies.

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. In the **Proxy** section of the Proxy Service tab, change the number in the **Adjusted Proxies** column and click **Apply** in the **Adjustment Plan** column.
6. In the Configure Proxy Resources dialog box, you can select **Migrate Immediately** to apply the changes. You can also select **Next Maintenance Period** to set a maintenance window for the change to take effect. Click **OK**.

View the monitoring data of dedicated proxies

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. Click the **Monitoring Data** tab.
6. Select a time range and view the **CPU Utilization (%)** metric within that time range.

Disable the dedicated proxy feature

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.

4. In the left-side navigation pane, click **Database Proxy**.
5. In the upper-right corner of the page, click **Disable Proxy Service**.
6. Click **OK**.

9.1.8.2. Short-lived connection optimization

This topic describes the short-lived connection optimization feature provided by ApsaraDB RDS for MySQL in its dedicated proxy feature.

Prerequisites

- The ApsaraDB RDS instance runs one of the following MySQL versions and RDS editions:
  - MySQL 5.7 on RDS Enterprise Edition
  - MySQL 5.7 on RDS High-availability Edition with local SSDs
- The database proxy is enabled for the instance. For more information, see Dedicated proxy.

Context

The short-lived connection optimization feature is used to reduce workloads on the ApsaraDB RDS instance caused by frequent short-lived connections. If a client is disconnected from a connection, the system determines whether the connection is idle. If the connection is idle, the proxy retains the connection in the connection pool for a short period. When your database client initiates a request to establish a connection again, the dedicated proxy matches the request with the idle connections that are retained in the session connection pool. The matching is implemented based on the values of the user, clientip, and dbname fields in the request. If the dedicated proxy finds an idle connection that matches the request, it uses the matched idle connection. If the dedicated proxy does not find an idle connection that matches the request, it establishes a new connection. If no idle connection is matched, a new connection is established with the database. This reduces the overheads of database connections.

> **Note** The short-lived connection optimization feature does not reduce concurrent connections with the database. It reduces the frequency to establish connections between the application and database and workloads of the primary MySQL thread. This improves efficiency to process business requests. However, idle connections in the connection pool still occupy the database threads for a short period of time.

Precautions

You cannot configure different permissions for the same account with different source IP addresses. Otherwise, errors may occur when connections in the connection pool are reused. For example, if the user account has permissions on database_a when its source IP address is 192.168.1.1 but does not have permissions on database_a when its source IP address is 192.168.1.2, the short-lived connection optimization feature may encounter permission errors.

Enable short-lived connection optimization

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. On the **Proxy Service** tab, click **Enable** to the right of **Short-Lived Connection Optimization**.

9.1.8.3. Transaction splitting

This topic describes the transaction splitting feature provided by the database proxy of ApsaraDB RDS. This feature identifies and distributes read requests initiated before write requests within a transaction to read-only instances. This reduces workloads on the primary instance.

Prerequisites

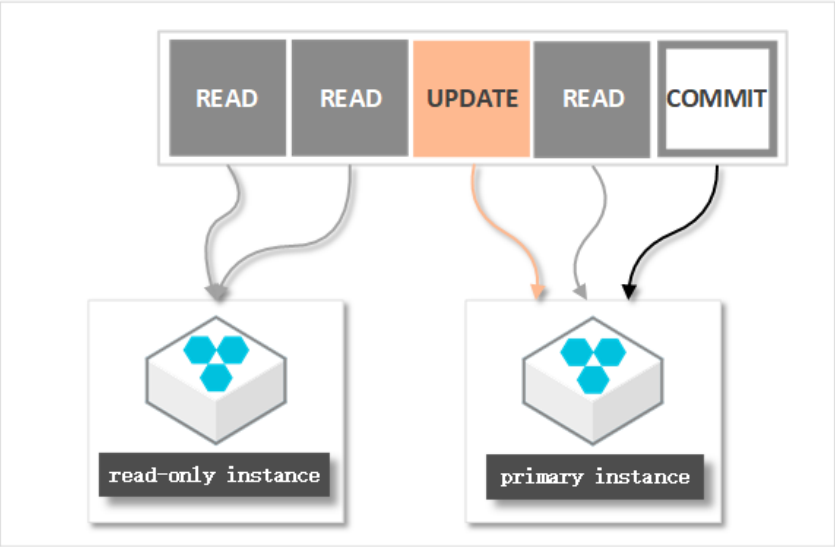
- The primary RDS instance runs one of the following MySQL versions and RDS editions:
  - MySQL 5.7 on RDS Enterprise Edition
  - MySQL 5.7 on RDS High-availability Edition with local SSDs
- The database proxy is enabled for the instance. For more information, see Dedicated proxy.

Context

By default, the dedicated proxy sends all requests in transactions to the primary instance to ensure the correctness of the transactions. If the framework used encapsulates all requests in transactions, the primary instance becomes heavily loaded. In this case, you can enable the transaction splitting feature.

When transaction splitting is enabled and the default isolation level READ COMMITTED is used, the ApsaraDB RDS instance starts a transaction only for write requests when autocommit is disabled (set autocommit=0). Read requests that arrive before the transaction is started are distributed to read-only instances by the load balancer.

> **Note**
> - Explicit transactions do not support splitting, such as transactions started by using the BEGIN or START statement.
> - After transaction splitting is enabled, global consistency cannot be ensured. If your business requires global consistency, we recommend that you evaluate whether transaction splitting suits your business requirements.



Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. On the **Proxy Service** tab, click **Enable** to the right of **Transaction Splitting**.

> Note
> - When you no longer need transaction splitting, you can click **Disable** to the right of **Transaction Splitting**.
> - The operation to enable or disable transaction splitting takes effect only on new connections.

9.1.8.4. Read/write splitting

9.1.8.4.1. Enable read/write splitting

This topic describes the read/write splitting feature of ApsaraDB RDS for MySQL in its dedicated proxy feature and how to enable this feature.

Prerequisites

- The ApsaraDB RDS instance runs one of the following MySQL versions and RDS editions:
  - MySQL 5.7 on RDS Enterprise Edition
  - MySQL 5.7 on RDS High-availability Edition
- You have enabled the database proxy or dedicated database proxy. For more information, see Enable the dedicated proxy feature.
- At least one read-only instance is created. For more information about how to create a read-only instance, see Create a read-only instance.

Context

If your primary instance needs to process a large number of read requests but only a small number of write requests, you can create one or more read-only instances to offload read requests from your primary instance. This ensures service stability. For more information, see Create a read-only instance.

After you create read-only instances, you can enable read/write splitting. In this case, a read/write splitting endpoint is provided. After you add the endpoint to your application, write requests are routed to the primary instance and read requests are routed to the read-only instances.

Differences between the read/write splitting endpoint and the internal and public endpoints

After you enable read/write splitting and add the read/write splitting endpoint to your application, all requests are routed to this endpoint, and then to the primary and read-only instances based on the request types and the read weights of these instances.

If the internal or public endpoint of the primary instance is added to your application, all requests are routed to the primary instance. In this case, you must add the endpoints and read weights of the primary and read-only instances to your application to implement read/write splitting.

Logic to route requests

- The following requests are routed only to the primary instance:
  - Data manipulation language (DML) statements, which are INSERT, UPDATE, DELETE, and SELECT FOR UPDATE
  - All data definition language (DDL) statements used to perform operations such as creating databases or tables, deleting databases or tables, and changing schemas or permissions
  - All requests that are encapsulated in transactions
  - Requests for user-defined functions
  - Requests for stored procedures
  - Requests for EXECUTE statements

  - Requests for multi-statements
  - Requests that involve temporary tables
  - Requests for SELECT last_insert_id() statements
  - All requests to query or modify user environment variables
  - Requests for SHOW PROCESSLIST statements
  - All requests for KILL statements in SQL (Note: These statements are not KILL commands in Linux.)
- The following requests are routed to the primary instance or its read-only instances:
  - Read requests that are not encapsulated in transactions
  - Requests for COM_STMT_EXECUTE statements
- The following requests are routed to all the primary and read-only instances:
  - All requests to modify system environment variables
  - Requests for USE statements
  - Requests for COM_STMT_PREPARE statements
  - Requests for COM_CHANGE_USER, COM_QUIT, and COM_SET_OPTION statements

Benefits

- Easier maintenance by using a unified endpoint

  If you do not enable the read/write splitting feature, you must add the endpoints of the primary and read-only instances to your application. After you add the endpoints, your database system routes write requests to the primary instance and read requests to the read-only instances.

  If you enable the read/write splitting feature, the endpoint of the dedicated proxy is used to implement read/write splitting. After your application is connected to this endpoint, your database system routes read and write requests to the primary and read-only instances based on the read weights of these instances. This reduces maintenance costs.

  In addition, you can scale up the read capability of your database system by creating read-only instances. This way, you do not need to modify the configuration data on your application.
- Higher performance and lower maintenance cost by using a native RDS link

  If you build a separate proxy layer on the cloud to implement read/write splitting, statements are parsed and forwarded by a number of components before they reach your database system. This increases response latency. The read/write splitting feature provided by ApsaraDB RDS shortens response latency, increases processing speed, and reduces maintenance costs.
- Ideal in various use scenarios based on configurable read weights and thresholds

  You can specify the read weights of the primary and read-only instances. You can also specify a latency threshold for each read-only instance.
- Highly available with instance-level health check

  The read/write splitting module actively performs health checks on the primary and read-only instances. If an instance breaks down or its latency exceeds the specified threshold, the read/write splitting module stops routing requests to the instance and redirects the requests that were destined for the instance to other healthy instances. Health checks on instances ensure service availability in the event of faults on a single read-only instance. After the faulty instance is recovered, the read/write splitting module resumes routing read requests to it.

  > **Note** To avoid single points of failure (SPOFs), we recommend that you create at least two read-only instances.

Precautions

- A network interruption may occur while the specifications of the primary instance or its read-only instances are being changed.
- After you create a read-only instance, only the requests over new connections can be routed to the read-only instance.
- The endpoint of the dedicated proxy does not support SSL encryption.
- The endpoint of the dedicated proxy does not support compression.
- If the endpoint of the dedicated proxy is used for connection, all the requests encapsulated in transactions are routed to the primary instance.
- If the endpoint of the dedicated proxy is used for read/write splitting, the read consistency of the requests that are not encapsulated in transactions cannot be ensured. If you require the read consistency, we recommend that you encapsulate the requests in transactions.
- If the endpoint of the dedicated proxy is used for connection, the `SHOW PROCESSLIST` statement combines the results from the primary and read-only instances and returns a result set.
- If short-lived connection optimization is enabled, the `SHOW PROCESSLIST` statement may return idle connections.
- If you execute multi-statements or stored procedures, read/write splitting is disabled and all the subsequent requests over the current connection are routed to the primary instance. To enable read/write splitting again, you must terminate the current connection and establish a new one.
- The `/*FORCE_MASTER*/` and `/*FORCE_SLAVE*/` hints are supported. However, requests that contain hints have higher route priorities. These requests are not constrained by consistency or transaction limits. You must check whether these hints are suitable for your business before you use them. A hint cannot contain statements that change environment variables. An example is `/*FORCE_SLAVE*/ set names utf8;`. Otherwise, an error may occur in the subsequent procedure.

Prerequisites

A read-only instance is created for the primary instance. For more information, see Create a read-only instance.

Enable read/write splitting

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. On the **Read/Write Splitting** tab, click **Enable now**.
6. Configure the following parameters.

Parameter	Description
Latency Threshold	The maximum latency that is allowed for data replication from the primary instance to its read-only instances. If the replication latency for a read-only instance exceeds the specified threshold, the read/write splitting module stops routing read requests to the read-only instance. This applies even if the read-only instance has a high read weight. Valid values: 0 to 7200. Unit: seconds. The read-only instances may replicate data from the primary instance at a certain latency due to SQL statement execution limits. We recommend that you set this parameter to a value that is greater than or equal to 30.

Parameter	Description
Read Weight Distribution	The read weight of each instance in your database system. A higher read weight indicates more read requests to process. For example, the primary instance is attached with three read-only instances, and the read weights of the primary and read-only instances are 0, 100, 200, and 200. In this situation, the primary instance only processes write requests, and the three read-only instances process all of the read requests at the 1:2:2 ratio. ○ **Automatic Distribution**: Your database system assigns a read weight to each instance based on the instance specifications. After you create a read-only instance, your database system assigns a read weight to the read-only instance and adds the read-only instance to the read/write splitting link. ○ **Customized Distribution**: You must manually specify the read weight of each instance. Valid values: 0 to 10000. After you create a read-only instance, its read weight is set to 0, which is the default value. You must manually specify the read weight of the read-only instance.

7. Click **OK**.

9.1.8.4.2. Configure read/write splitting

This topic describes how to configure the latency threshold and specify read weights for an ApsaraDB RDS instance in the ApsaraDB RDS console.

Prerequisites

- The ApsaraDB RDS instance runs one of the following MySQL versions and RDS editions:
  - MySQL 5.6
  - MySQL 5.7 on RDS Enterprise Edition
  - MySQL 5.7 on RDS High-availability Edition with local SSDs
- Read/write splitting is enabled. For more information, see Enable read/write splitting.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. On the **Read/Write Splitting** tab, click **Configure Read/Write Splitting**.
6. Configure the following parameters.

Parameter	Description
Latency Threshold	The maximum latency that is allowed for data replication from the primary instance to its read-only instances. If the latency of data replication to a read-only instance exceeds the specified threshold, ApsaraDB RDS stops routing read requests to the instance. This applies even if the instance has a high read weight. Valid values: 0 to 7200. Unit: seconds. The read-only instances may replicate data from the primary instance at a specific latency due to SQL statement execution limits. We recommend that you set this parameter to a value that is greater than or equal to 30.

Parameter	Description
Read Weight Distribution	The read weight of each instance in your database system. A higher read weight indicates more read requests to process. For example, assume that your primary instance is attached with three read-only instances, and the read weights of the primary and read-only instances are 0, 100, 200, and 200. In this case, your primary instance processes only write requests, and the three read-only instances process all of the read requests at the 1:2:2 ratio. ○ **Automatic Distribution**: Your database system assigns a read weight to each instance based on the instance specifications. After you create a read-only instance, your database system assigns a read weight to the read-only instance and adds the read-only instance to the read/write splitting link. ○ **Customized Distribution**: You must manually specify the read weight of each instance. Valid values: 0 to 10000. After you create a read-only instance, ApsaraDB RDS sets the read weight of the read-only instance to 0. You must manually modify the read weight of the created read-only instance.

7. Click **OK**.

9.1.8.4.3. Disable read/write splitting

This topic describes how to disable the read/write splitting feature of an ApsaraDB RDS instance in the ApsaraDB RDS console.

Prerequisites

- The ApsaraDB RDS instance runs one of the following MySQL versions and RDS editions:
  - MySQL 5.6
  - MySQL 5.7 on RDS Enterprise Edition
  - MySQL 5.7 on RDS High-availability Edition with local SSDs
- Read/write splitting is enabled. For more information, see Enable read/write splitting.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. On the **Read/Write Splitting** tab, click **Disable Read/Write Splitting**.
6. Click **OK**.

9.1.9. Monitoring and alerts

9.1.9.1. View resource and engine monitoring data

The ApsaraDB RDS console provides a variety of performance metrics to monitor the status of your instances.

Prerequisites

The ApsaraDB RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 5.6
- MySQL 5.7 on RDS High-availability Edition with local SSDs

- MySQL 5.7 on RDS High-availability Edition with standard SSDs

Procedure

1. For more information, see Log on to the ApsaraDB RDS console.
2. Find an instance and click the instance ID.
3. In the left-side navigation pane, click **Monitoring and Alerts**.
4. On the **Monitoring and Alerts** page, select **Resource Monitoring** or **Engine Monitoring**, and select a time range to view the corresponding monitoring data. The following table describes the metrics.

Category	Metric	Description
Resource Monitoring	**Disk Space (MB)**	The disk space usage of the instance. It consists of the following items: ○ Instance size ○ Data usage ○ Log size ○ Temporary file size ○ Other system file size Unit: MB.
	IOPS (Input/Output Operations per Second)	The number of input/output operations per second (IOPS) of the instance.
	Total Connections	The number of active connections to the instance and the total number of connections to the instance.
	CPU Utilization and Memory Usage (%)	The CPU utilization and memory usage of the instance. These metrics do not include the CPU utilization and memory usage for the operating system.
	Network Traffic (KB)	The inbound and outbound traffic of the instance per second. Unit: KB.
	Transactions per Second (TPS)/Queries per Second (QPS)	The average number of transactions per second and the average number of SQL statements executed per second.
	InnoDB Buffer Pool Read Hit Ratio, Usage Ratio, and Dirty Block Ratio (%)	The read hit ratio, usage ratio, and dirty block ratio of the InnoDB buffer pool.
	InnoDB Read/Write Volume (KB)	The amount of data that InnoDB reads and writes per second. Unit: KB.
	InnoDB Buffer Pool Read/Write Frequency	The number of read and write operations that InnoDB performs per second.
	InnoDB Log Read/Write/fsync	The average frequency of physical writes to log files per second by InnoDB, the log write request frequency, and the average frequency of fsync writes to log files.

Category	Metric	Description
Engine Monitoring	**Temporary Tables Automatically Created on Hard Disk when MySQL Statements Are Executed**	The number of temporary tables that are automatically created on the hard disk when the database executes SQL statements.
	MySQL_COMDML	The number of SQL statements that the database executes per second. The following SQL statements are included: ○ Insert ○ Delete ○ Insert_Select ○ Replace ○ Replace_Select ○ Select ○ Update
	MySQL_RowDML	The numbers of operations that InnoDB performs per second. The following items are included: ○ The number of physical writes to log files per second. ○ The number of rows that are read, updated, deleted, and inserted from InnoDB tables per second.
	MyISAM Read/Write Frequency	The numbers of operations that MyISAM performs per second. The following items are included: ○ The number of MyISAM reads and writes from the buffer pool per second. ○ The number of MyISAM reads and writes from the hard disk per second.
	MyISAM Key Buffer Read/Write/Usage Ratio (%)	The read hit ratio, write hit ratio, and usage of the MyISAM key buffer per second.

9.1.9.2. Set a monitoring frequency

The ApsaraDB RDS console provides a variety of performance metrics for which you can set a monitoring frequency.

Prerequisites

The ApsaraDB RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 5.6
- MySQL 5.7 on RDS High-availability Edition with local SSDs
- MySQL 5.7 on RDS High-availability Edition with standard SSDs

Context

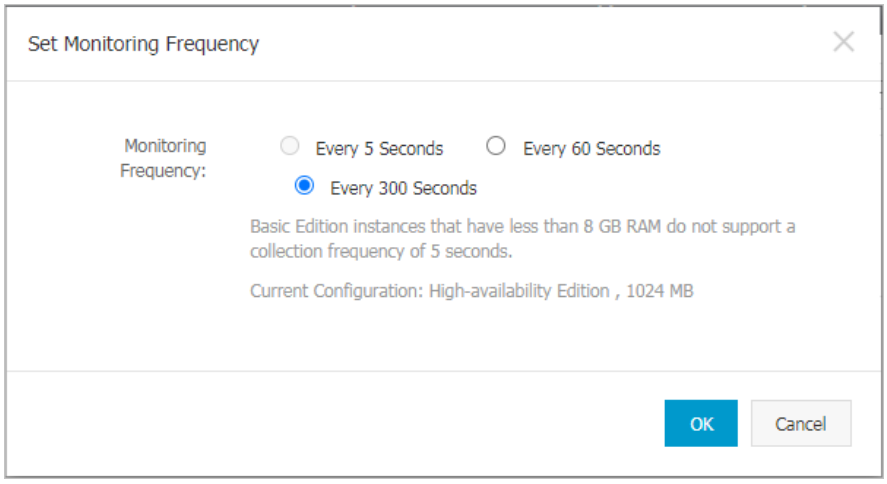
ApsaraDB RDS provides the following monitoring frequencies:

- Every 5 seconds for the first seven days. After the seventh day, performance metrics are monitored every minute.

- Every 60 seconds.
- Every 300 seconds.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. On the **Monitoring** tab, click **Set Monitoring Frequency**.
6. In the **Set Monitoring Frequency** dialog box, select a new monitoring frequency.



> **Note** If the RDS instance runs the RDS Basic Edition or its memory capacity is less than 8 GB, the **Every 5 Seconds** monitoring frequency is not supported.

7. Click **OK**.

9.1.10. Data security

9.1.10.1. Configure a whitelist

To ensure database security and reliability, you must modify the whitelist of an ApsaraDB RDS instance before you enable the instance. You must add the IP addresses or CIDR blocks that are used for database access to the whitelist.

Context

The whitelist improves the access security of your ApsaraDB RDS instance. We recommend that you maintain the whitelist on a regular basis. The whitelist configuration process does not affect the normal operations of the ApsaraDB RDS instance.

To configure a whitelist, perform the following operations:

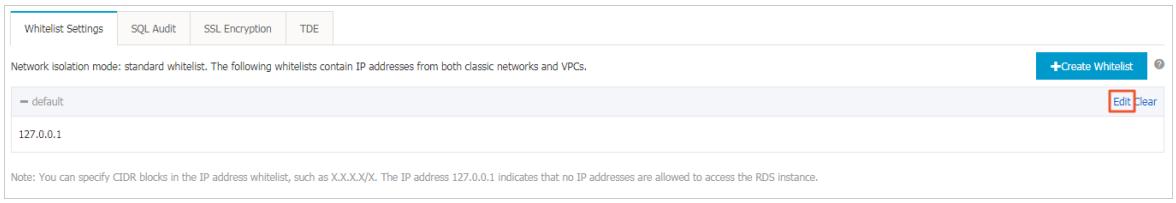
- Configure a whitelist: Add IP addresses to allow them to connect to the ApsaraDB RDS instance.
- Configure an ECS security group: Add an ECS security group for the ApsaraDB RDS instance to allow ECS instances in the group to connect to the ApsaraDB RDS instance.

Precautions

- The default whitelist can be modified or cleared, but cannot be deleted.
- You can add up to 1,000 IP addresses or CIDR blocks to a whitelist. If you want to add a large number of IP addresses, we recommend that you merge them into CIDR blocks, such as 192.168.1.0/24.

Configure a standard IP address whitelist

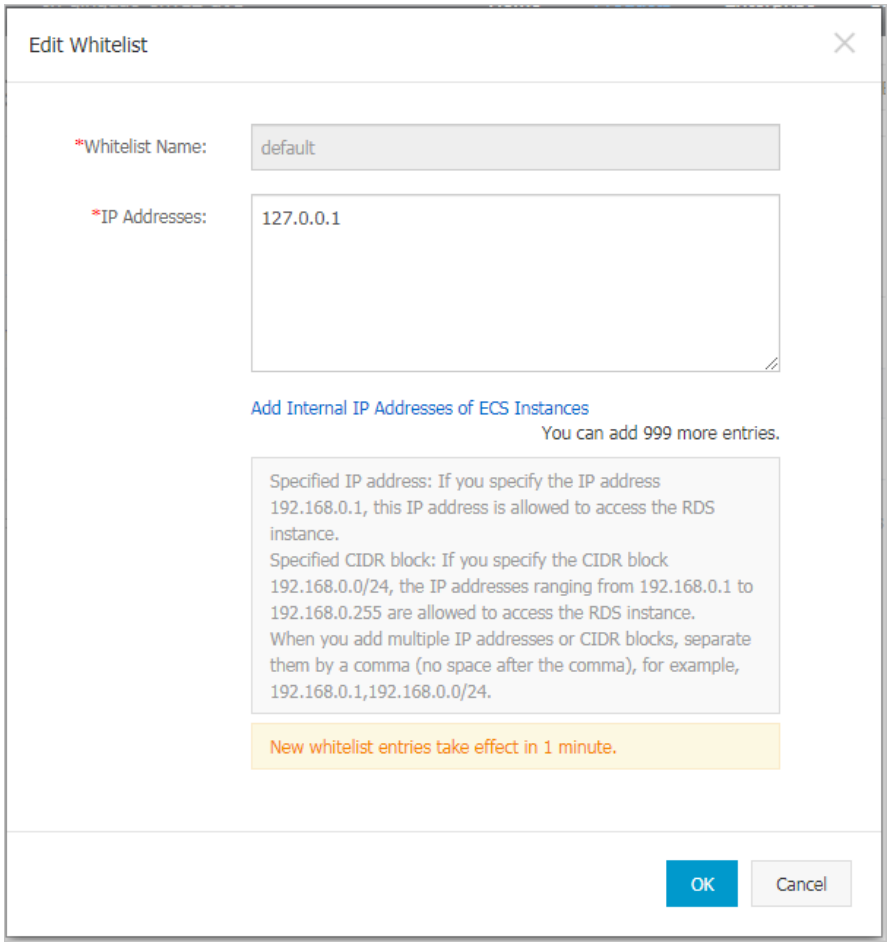
1. For more information, see Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find an instance. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
3. In the left-side navigation pane, click **Data Security**.
4. On the **Whitelist Settings** tab, click **Edit** corresponding to the **default** whitelist.



> **Note**
> - If you want to connect an ECS instance to an ApsaraDB RDS instance by using an internal endpoint, you must make sure that the two instances are in the same region and have the same network type. Otherwise, the connection fails.
> - You can also click **Create Whitelist** to create a new whitelist.

5. In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks that are allowed to access your ApsaraDB RDS instance, and then click **OK**.
   - If you add the CIDR block 10.10.10.0/24, all IP addresses in the 10.10.10.X format are allowed to access the ApsaraDB RDS instance.
   - If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
   - If you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all of the ECS instances that are created in your Alibaba Cloud account appear. Then, you can select the required IP addresses and add them to the whitelist.

> **Note** If you add a new IP address or CIDR block to the **default** whitelist, the default address 127.0.0.1 is deleted.



9.1.10.2. Configure SSL encryption

This topic describes how to enhance endpoint security. You can enable secure sockets layer (SSL) encryption and install SSL certificates issued by certificate authorities (CAs) on the required application services. SSL is used on the transport layer to encrypt network connections and enhance the security and integrity of communication data. However, SSL also increases the response time.

Prerequisites

The ApsaraDB RDS instance runs one of the following MySQL versions and RDS editions:

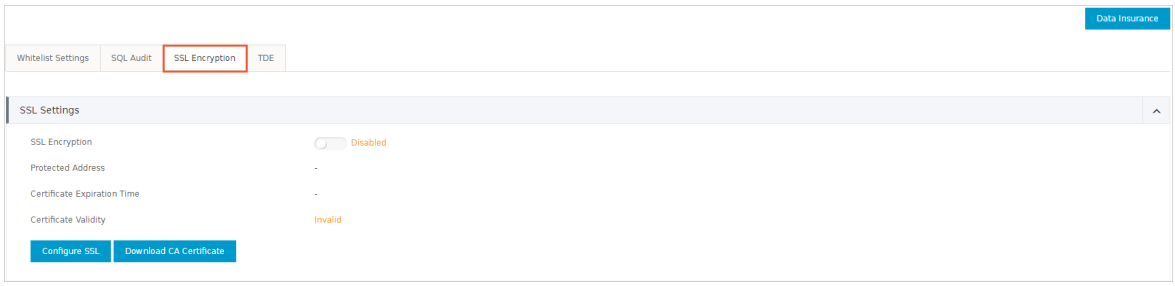
- MySQL 5.6
- MySQL 5.7 on RDS High-availability Edition with local SSDs
- MySQL 5.7 on RDS High-availability Edition with standard SSDs

Precautions

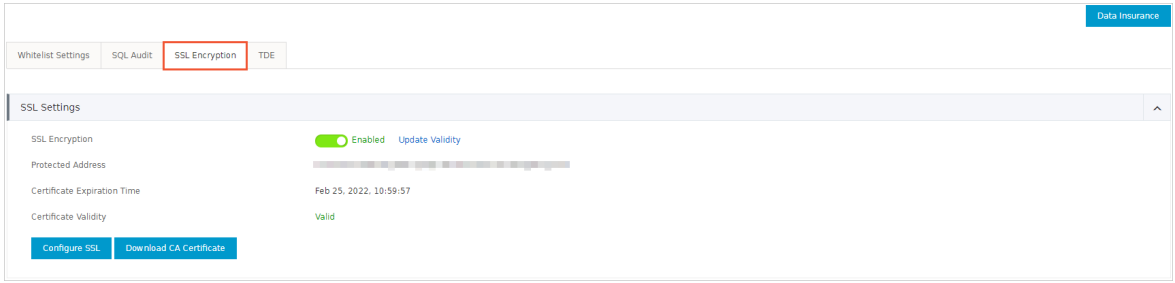
- An SSL CA certificate is valid for one year. You must update the validity period of the certificate and then download and configure the certificate again. Otherwise, clients that use encrypted connections cannot connect to the instance.
- SSL encryption may cause a significant increase in CPU utilization. We recommend that you enable SSL encryption only when you want to encrypt connections from the Internet. In most cases, connections that use an internal endpoint do not require SSL encryption.
- Read/write splitting endpoints do not support SSL encryption.
- If you disable SSL encryption, the ApsaraDB RDS instance restarts. Proceed with caution.

Enable SSL encryption

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SSL Encryption** tab.



6. In the SSL Settings section, turn on **SSL Encryption**.
7. In the **Configure SSL** dialog box, select the endpoint for which you want to enable SSL encryption and click **OK**.
8. Click **Download CA Certificate** to download the SSL CA certificate files in a compressed package.



The downloaded package contains the following files:

- P7B file: used to import CA certificates to the Windows system.
- PEM file: used to import CA certificates to other operating systems or applications.
- JKS file: the Java truststore file. The password is apsaradb. It is used to import the CA certificate chain to Java programs.

> **Note** When the JKS file is used in Java, you must modify the default JDK security configuration in JDK7 and JDK8. Open the `/jre/lib/security/java.security` file on the host where your application resides, and modify the following configurations:
>
> ```
> jdk.tls.disabledAlgorithms=SSLv3, RC4, DH keySize < 224
> jdk.certpath.disabledAlgorithms=MD2, RSA keySize < 1024
> ```
>
> If you do not modify the JDK security configuration, the following error is reported. Similar errors are also caused by the Java security configuration.
>
> ```
> javax.net.ssl.SSLHandshakeException: DHPublicKey does not comply to algorithm constraints
> ```

Configure an SSL CA certificate

After you enable SSL encryption, configure the SSL CA certificate on your application or client before they can connect to the instance. The following section shows how to configure an SSL CA certificate. MySQL Workbench and Navicat are used in the example. For more information, see the instructions for the other applications or clients.

Configure a certificate on MySQL Workbench

1. Start MySQL Workbench.
2. Choose **Database > Manage Connections**.
3. Enable **Use SSL** and import the SSL CA certificate files.

Configure a certificate on Navicat

1. Start Navicat.
2. Right-click the database and select **Edit Connection**.
3. Click the **SSL** tab. Select the path of the SSL CA certificate file in the PEM format, as shown in the following figure.
4. Click **OK**.

   > **Note** If the `connection is being used` error is reported, the previous session is still connected. Restart Navicat.

5. Double-click the database to test whether the database is connected.

Update the validity period of an SSL CA certificate

> **Note** **Update Validity** causes the ApsaraDB RDS instance to restart. Proceed with caution.

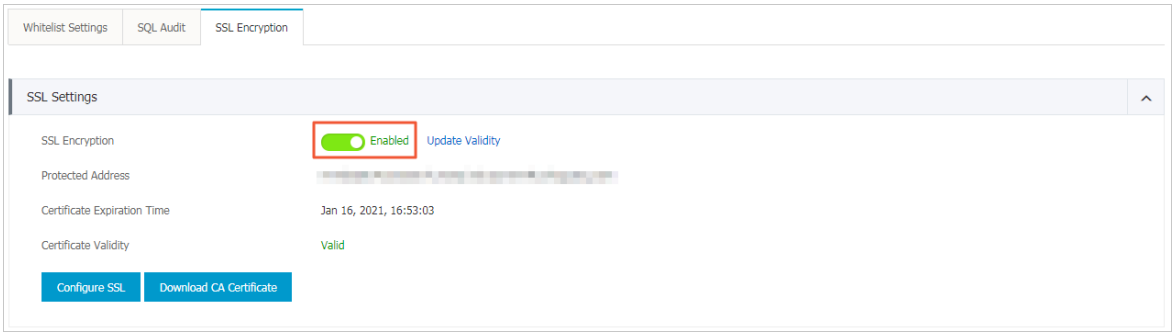
1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SSL Encryption** tab.
6. Click **Update Validity**.

Disable SSL encryption

> **Note**
> - If you disable SSL encryption, the ApsaraDB RDS instance restarts. To reduce the impact on your business, the system triggers a primary/secondary switchover. We recommend that you disable SSL encryption during off-peak hours.
> - After you disable SSL encryption, access performance increases, but security decreases. We recommend that you disable SSL encryption only in secure environments.

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SSL Encryption** tab.

6. Turn off **SSL Encryption**. In the message that appears, click **OK**.



9.1.10.3. Configure TDE

This topic describes how to configure Transparent Data Encryption (TDE) for your ApsaraDB RDS for MySQL instance. TDE encrypts and decrypts data files in real time. It encrypts data files when they are written to disks, and decrypts data files when they are loaded to the memory from disks. TDE does not increase the size of data files. You can use TDE without the need to make changes to applications.

Prerequisites

- Your ApsaraDB RDS instance runs one of the following MySQL versions and RDS editions:
  - MySQL 5.7 on RDS High-availability Edition with local SSDs
  - MySQL 5.6
- Key Management Service (KMS) is activated. If KMS is not activated, you can activate it when you enable TDE.

Context

The key used for TDE is created and managed by KMS. ApsaraDB RDS does not provide the key or certificates that are required for encryption. For specific zones, you can use the keys that are automatically generated by Apsara Stack or use your own key materials to generate data keys, and then authorize your ApsaraDB RDS for MySQL instance to use these keys.

Precautions

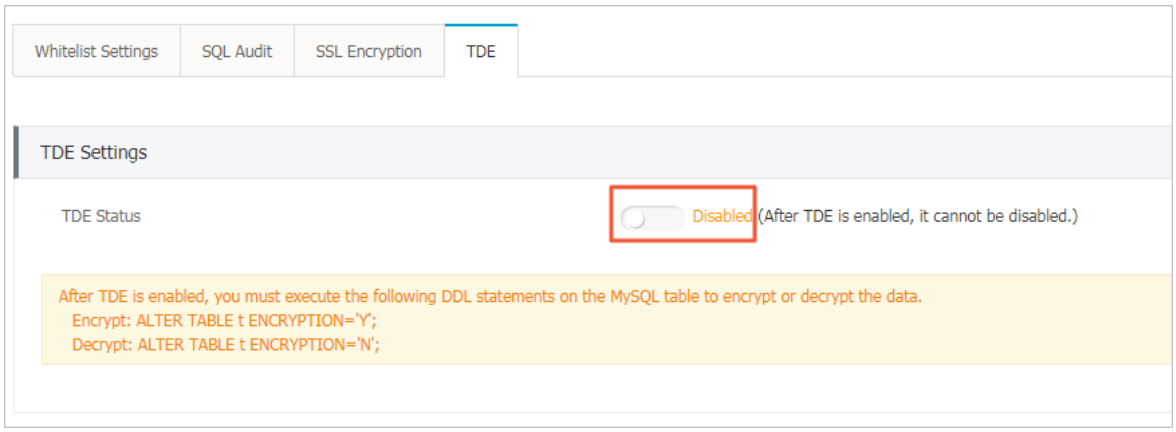
- When TDE is being enabled, your ApsaraDB RDS for MySQL instance is restarted and all its connections are terminated. Make appropriate service arrangements before you enable TDE. Proceed with caution.
- You cannot disable TDE after it is enabled.
- You cannot change the key used for encryption after TDE is enabled.
- If you want to restore the data to your computer after TDE is enabled, you must decrypt data on your ApsaraDB RDS for MySQL instance. For more information, see the Decrypt data section in this topic.
- After TDE is enabled, CPU utilization significantly increases.
- If you use an existing custom key for encryption, take note of the following items:
  - If you disable a key, set a key deletion plan, or delete the key materials, the key becomes unavailable.
  - After you revoke the key that is authorized for an ApsaraDB RDS for MySQL instance, the instance becomes unavailable after it is restarted.
  - You must use an Apsara Stack account or an account that has the AliyunSTSAssumeRoleAccess permission.

> **Note** For more information, see topics about key management in *Key Management Service User Guide*

Use a key that is automatically generated by Apsara Stack

1. Log on to the ApsaraDB for RDS console.

2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **TDE** tab. Then, turn on **TDE Status**.

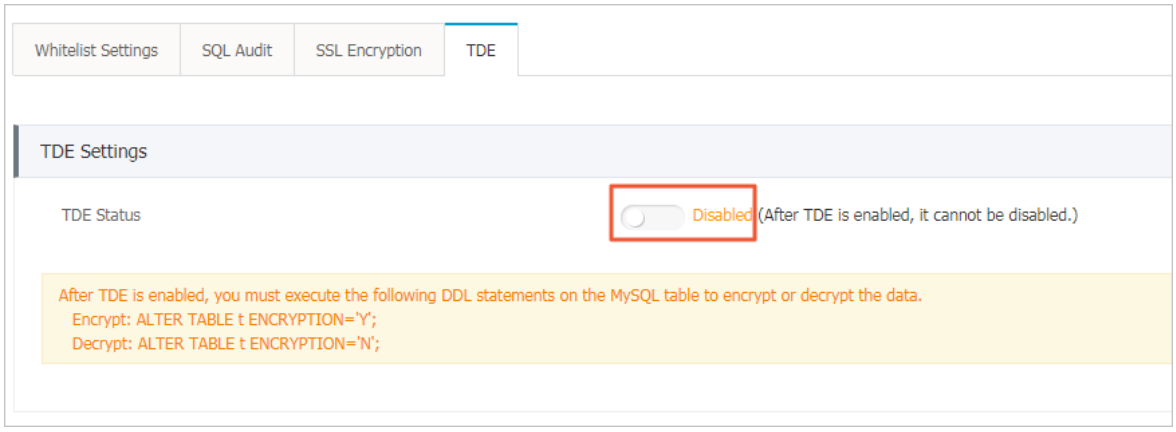


6. In the dialog box that appears, select **Use an Automatically Generated Key** and click **OK**.

> **Note** If the instance runs MySQL 5.7 on RDS High-availability Edition, you can select one of the following encryption methods:
>
> - SM4 Encryption
> - AES_256_CBC Encryption

Use an existing custom key

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **TDE** tab. Then, turn on **TDE Status**.



6. In the dialog box that appears, select **Use an Existing Custom Key** and click **OK**.

> **Note** If you do not have a custom key, click **create a key** to go to the KMS console and import the key materials. For more information, see Create a key in *Key Management Service User Guide*.

Encrypt a table

Log on to the database and execute the following statement to encrypt a table:

- MySQL 5.6

```
alter table <tablename> engine=innodb,block_format=encrypted;
```

- MySQL 5.7

```
alter table <tablename> engine=innodb,encryption='Y';
```

Decrypt a table

Execute the following statement to decrypt a table that is encrypted by using TDE:

- MySQL 5.6

```
alter table <tablename> engine=innodb,block_format=default;
```

- MySQL 5.7

```
alter table <tablename> engine=innodb,encryption='N';
```

FAQ

- Q: Can common database tools such as Navicat be used after TDE is enabled?

  A: Yes, after you enable TDE, you can still use common database tools, such as Navicat.

- Q: Why is data still in plaintext after it is encrypted?

  A: After you enable TDE, your data is stored in ciphertext. However, when the data is queried, it is decrypted and loaded to the memory in plaintext. TDE encrypts backup files to prevent data leaks. The encrypted backup files cannot be used to restore data to your computer. If you want to restore these backup files to your computer, you must decrypt them. For more information, see the Decrypt a table section in this topic.

9.1.10.4. SQL audit

You can use the SQL audit feature to audit SQL executions and check the details. SQL audit does not affect instance performance.

Context

> **Note** You cannot view the logs that are generated before you enable SQL audit.

You can view the incremental data of your ApsaraDB RDS for MySQL instance in SQL audit logs or binlogs. However, these two methods differ in the following aspects:

- SQL audit logs are similar to audit logs in MySQL and record all DML and DDL operations by using network protocol analysis. SQL audit does not parse the actual parameter values. Therefore, a small amount of information may be lost if a large number of SQL statements are executed to query data. The incremental data obtained by using this method may be inaccurate.
- Binlogs record all add, delete, and modify operations and the incremental data used for data restoration. Binlogs are temporarily stored in your ApsaraDB RDS instance after they are generated. The system transfers full binlog files to OSS on a regular basis. OSS then stores the files for seven days. However, a binlog file cannot be transferred if data is being written to it. Such binlog files cannot be uploaded to OSS after you click **Upload**

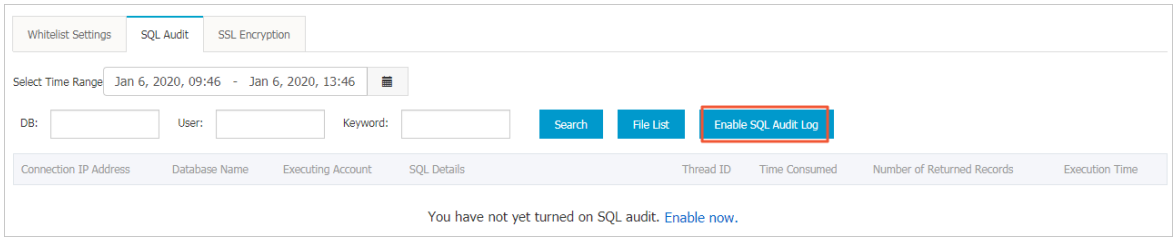
Binlogs on the **Backup and Restoration** page. Binlogs are not generated in real time, but you can obtain accurate incremental data from them.

Precautions

- SQL audit is disabled by default. SQL audit does not affect instance performance.
- SQL audit logs are retained for 30 days.
- Log files exported from SQL audit are retained for two days. The system clears files that are retained for more than two days.

Enable SQL audit

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SQL Audit** tab.



6. Click **Enable SQL Audit**.
7. In the message that appears, click **Confirm**.

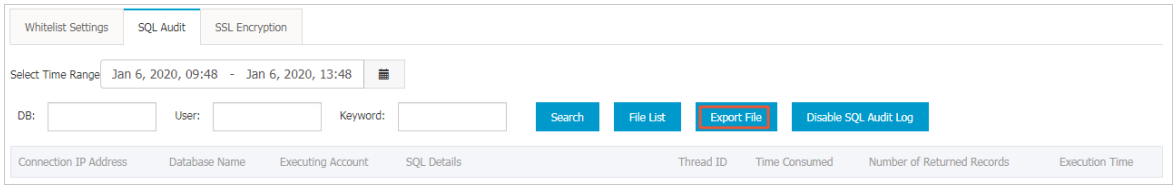
   After SQL audit is enabled, you can query SQL information based on conditions such as the time range, database, user, and keyword.

Disable SQL audit

> **Note** If SQL audit is disabled, all SQL audit logs are deleted. We recommend that you export and store audit logs to your computer before you disable SQL audit.

You can disable SQL audit to avoid charges when you do not need it. To disable SQL audit, perform the following operations:

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SQL Audit** tab.



6. Click **Export File** to export and store the SQL audit content to your computer.

7. After the file is exported, click **Disable SQL Audit**.
8. In the message that appears, click **Confirm**.

9.1.11. Service availability

9.1.11.1. Configure automatic or manual switchover

This topic describes how to automatically or manually switch over services between primary and secondary instances. After a switchover, the original primary instance becomes a secondary instance.

Context

- Automatic switchover: the default switchover mode. If the primary instance experiences a fault, your ApsaraDB RDS services are automatically switched over to the secondary instance.

  > **Note** You can click **Switch Primary/Secondary Instance** on the **Service Availability** page of an ApsaraDB RDS for MySQL instance with standard or enhanced SSDs to disable automatic switchover. This facilitates troubleshooting when errors occur on the primary instance.

- Manual switchover: allows you to manually switch over services between primary and secondary instances.

> **Note** Data is synchronized between the primary and secondary instances in real time. You can connect only to the primary instance. The secondary instances serve only as backups and do not allow external access.

Precautions

- Services may be disconnected during a switchover. Make sure that your applications are configured with automatic reconnection policies to avoid service disruptions.
- If the primary instance is attached with read-only instances, data on the read-only instances shows a latency of several minutes after a switchover. This is because it takes time to re-establish replication connections and synchronize incremental data.

Manually switch over services between primary and secondary instances

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Service Availability**.
5. Click **Switch Primary/Secondary Instance** on the right side of the page.

   > **Note** Services may be disconnected once or twice during the switchover. Make sure that your applications are configured with automatic reconnection policies to avoid service disruptions.

6. In the dialog box that appears, click **OK**.

FAQ

Q: Can I connect to secondary instances?

A: No, you cannot connect to secondary instances. You can connect only to primary instances. Secondary instances serve only as backups and do not allow external access.

9.1.11.2. Change the data replication mode

You can set the data replication mode between primary and secondary ApsaraDB RDS instances to improve database availability.

Prerequisites

Your ApsaraDB RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 5.6
- MySQL 5.7 on RDS High-availability Edition with local SSDs

Data replication modes

- Semi-synchronous

  After an update that is initialized by your application is complete on the primary instance, the log is synchronized to all the secondary instances. The transaction that is used to perform the update is considered committed after the secondary instances receive the log. Your database system does not need to wait for the log to be replayed.

  If the secondary instances are unavailable or a network exception occurs between the primary and secondary instances, semi-synchronous replication degrades to the Asynchronous mode.

- Asynchronous

  When your application initializes a request to add, delete, or modify data, the primary instance responds to your application immediately after it completes the operation. At the same time, the primary instance starts to asynchronously replicate data to its secondary instances. During asynchronous data replication, the unavailability of secondary instances does not affect the operations on the primary instance. Data remains consistent even if the primary instance is unavailable.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Service Availability**.
5. Click **Change Data Replication Mode** in the upper-right corner of the Availability Information section.
6. In the Change Data Replication Mode dialog box, select a data replication mode and click **OK**.

FAQ

Q: Which data replication mode is recommended?

A: You can select a data replication mode based on your business requirements. If you require quick responses, we recommend that you select the asynchronous mode. In other scenarios, you can select the semi-synchronous mode.

9.1.12. Database backup and restoration

9.1.12.1. Automatic backup

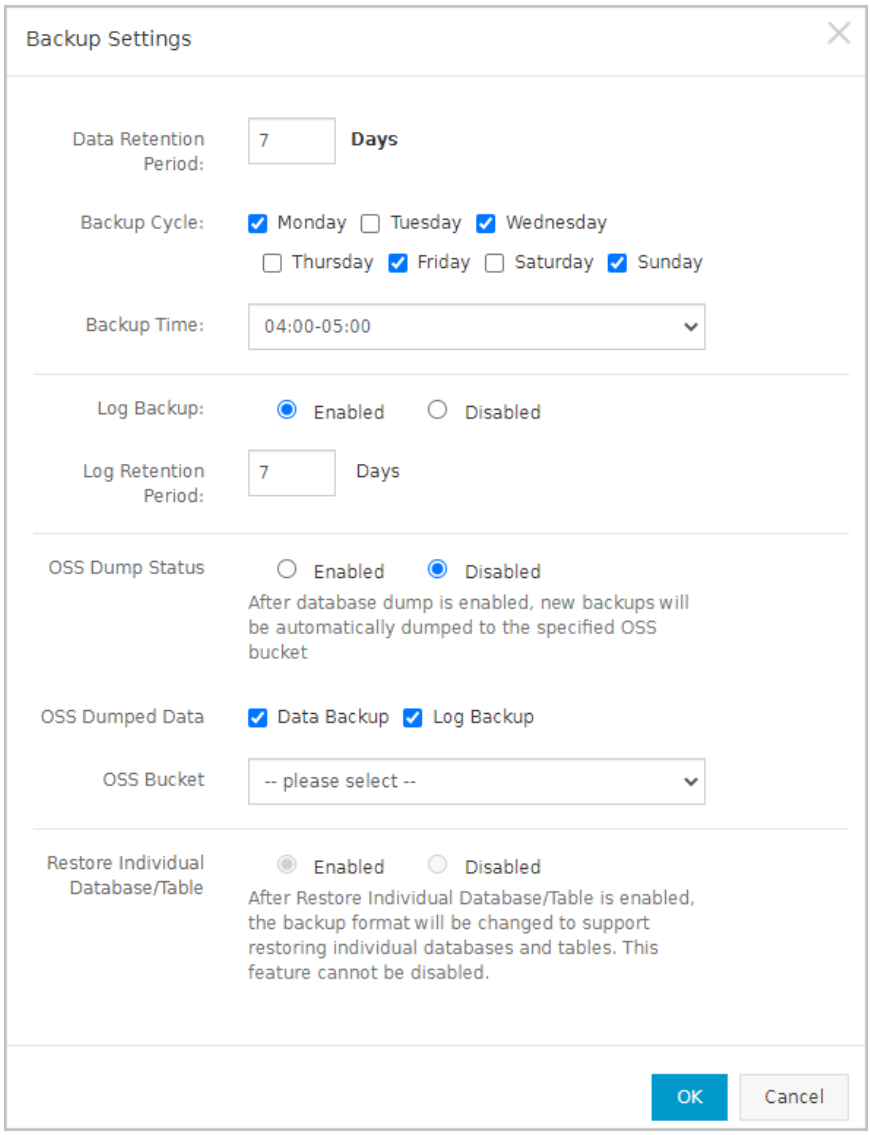
ApsaraDB RDS automatic backup supports full physical backups. ApsaraDB RDS automatically backs up data based on pre-configured policies. This topic describes how to configure a policy for automatic backup.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.

3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. Click the **Backup Settings** tab.
6. Click **Edit**.

> **Note** To ensure data security, the system compares the new backup cycle and time with the original settings, and selects the most recent time point to back up the data. Therefore, the next backup may still be performed based on the original backup cycle and time. For example, if the backup time is set to 19:00-20:00 every Wednesday and you modify the time to 19:00-20:00 every Thursday before 19:00 this Wednesday, the system still backs up data during 19:00-20:00 this Wednesday.



7. Configure the following parameters.

Parameter	Description
Data Retention Period	The number of days for which data backup files are retained. Valid values: 7 to 730. Default value: 7.

Parameter	Description
Backup Cycle	The backup cycle. You can select one or multiple days within a week.
Backup Time	A period of time within a day. Unit: hours. We recommend that you back up data during off-peak hours.
Log Backup	Specifies whether to enable log backup. **Notice** If you disable log backup, all the log backup files are deleted, and you cannot restore data to a saved point in time.
Log Retention Period	The number of days for which log backup files are retained. Valid values: 7 to 730. Default value: 7.
Restore Individual Database/Table	Specifies whether to enable restoration of individual databases or tables. You cannot disable this feature after it is enabled. **Note** Restoration of individual databases or tables can be enabled only on instances with **local SSDs**. For more information about this feature, see Restore individual databases and tables for an ApsaraDB RDS for MySQL instance.

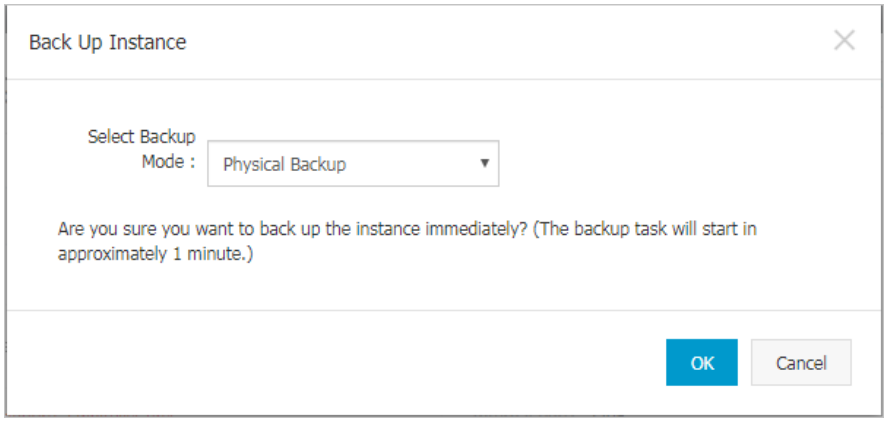
8. After you configure the preceding parameters, click **OK**.

9.1.12.2. Manual backup

Manual backup supports both full physical backups and full logical backups. This topic describes how to manually back up ApsaraDB RDS data.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. Click **Back Up Instance** in the upper-right corner.



5. Set the backup mode and backup policy, and click **OK**.

> **Note** Two backup methods are available:
>
> - Physical backup: directly backs up all files in all databases.
> - Logical backup: extracts data from the databases by using SQL statements and backs up the data in the text format. If you select logical backup, you must select a backup policy:
>   - Instance Backup: backs up the entire instance.
>   - Single-Database Backup: backs up one of the databases in the instance.

9.1.12.3. Restore individual databases and tables for an ApsaraDB RDS for MySQL instance

This topic describes how to restore the individual databases and tables that are accidentally deleted from an ApsaraDB RDS for MySQL instance. You can restore these databases and tables from backup files.

Prerequisites

- Your ApsaraDB RDS instance runs one of the following MySQL versions and RDS editions:
  - MySQL 5.7 on RDS High-availability Edition with local SSDs
  - MySQL 5.6 on RDS High-availability Edition
- The number of tables on the ApsaraDB RDS instance does not exceed 50,000.
- If you want to restore individual databases and tables of your ApsaraDB RDS instance to the same instance, the ApsaraDB RDS instance meets the following requirements:
  - The ApsaraDB RDS instance is in the Running state and is not locked.
  - The ApsaraDB RDS instance does not have ongoing migration tasks.
  - If you want to restore individual databases and tables to a point in time, the log backup feature is enabled.
  - If you want to restore an instance from a backup set, at least one backup set is available.

> **Note** For more information about how to restore databases at the instance level, see Restore data to a new instance (formerly known as cloning an instance).

Precautions

- If you restore individual databases and tables to the original ApsaraDB RDS instance, a primary/secondary switchover is triggered. This may cause a network interruption. Make sure that your application is configured to automatically reconnect to the original ApsaraDB RDS instance. If you restore individual databases and tables to a new ApsaraDB RDS instance, no primary/secondary switchover is triggered.
- The Restore Individual Database/Table feature restores only the selected tables. You must select all of the tables that you want to restore. The restoration fails in the following scenarios:
  - The selected tables are deleted during the specified period. The specified period spans from the point in time when the last backup set is generated to the point in time to which you want to restore the selected tables.
  - The restoration involves a table that you have not selected. For example, you selected Table B, but Table B was renamed from Table A before the specified point in time. In this case, the restoration fails because you did not select Table A.
- You can select a maximum of 50 databases or tables at a time.

Procedure

1. Log on to the ApsaraDB for RDS console.

2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. In the upper-right corner of the page, click **Restore Individual Database/Table**.

   > **Note** If the **Restore Individual Database/Table** button is not displayed, you can check whether all the requirements that are specified in the "Prerequisites" section of this topic are met.

6. Configure the following parameters.

Parameter	Description
Restore To	**Current Instance**: restores individual databases and tables to the original ApsaraDB RDS instance.
Restore Method	◦ **By Backup Set** ◦ **By Time**: restores data to a point in time within the specified log backup retention period. **Note** The **By Time** option appears only when the log backup feature is enabled.
Backup Set	Select the backup set from which you want to restore individual databases and tables. **Note** This parameter appears only when you set the **Restore Method** parameter to **By Backup Set**.
Restore Time	Select the point in time to which you want to restore individual databases and tables. **Note** This parameter appears only when you set the **Restore Method** parameter to **By Time**.
Databases and Tables to Restore	Select the databases or tables that you want to restore.
Selected Databases and Tables	◦ This section displays the selected databases and tables. You can specify new names for these databases and tables. ◦ This section also displays the total size of the selected databases and tables and the remaining storage space. Make sure that the remaining storage space is sufficient before the restoration.

7. Click **OK**.

FAQ

- After the backup file format is changed from TAR to xbstream, are the existing backup files in the TAR format still available?

  Yes, the original backup files in the TAR format are still available.

- Why does the Restore Individual Database/Table feature suddenly become unavailable?

Check whether the number of tables on your ApsaraDB RDS instance exceeds 50,000. If the number exceeds 50,000, the Restore Individual Database/Table feature is unavailable.

9.1.12.4. Download data and log backup files

This topic describes how to download unencrypted data and log backup files in the ApsaraDB RDS console to archive the files and restore data to an on-premises database.

Limits

Database engine	Download of data backup files	Download of binlog files
MySQL 5.6 on RDS High-availability Edition	Supported	Supported
MySQL 5.7 on RDS High-availability Edition with local SSDs	Supported	Supported
MySQL 5.7 on RDS High-availability Edition with standard SSDs	Not supported	Supported

Procedure

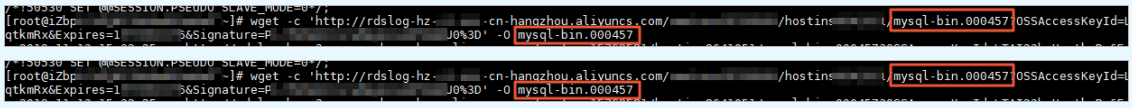
1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. Click the Data Backup or Log Backup tab.
   - To download data backup files, click the **Data Backup** tab.
   - To download log backup files, click the **Log Backup** tab.
6. Select a time range to which you want to restore the instance.
7. Find the data backup or log file you want to download, and click **Download** in the **Actions** column.

> **Note**
> - If the Download button is unavailable, see the Limits section in this topic.
> - If you want to use a data backup file to restore data, select the backup file that is the closest to the time for restoration.
> - If you want to use a log file to restore data to an on-premises database, take note of the following items:
>   - The Instance ID on the Log Backup tab is the same as the Instance No. on the Data Backup tab.
>   - The start time of the file is later than the start time of the specified time range and earlier than the point in time to which you want to restore data.

8. In the download message that appears, click **Download**.

Download method	Description
Download	Use a browser to download the backup file.
Copy Internal Endpoint	Copy the internal endpoint to download the file. If your ECS and ApsaraDB RDS instances reside within the same region, you can log on to the ECS instance and use the internal endpoint to download the file. This method is fast and secure.
Copy Public Endpoint	Copy the public endpoint to download the file. If you want to use other tools to download the file, use the public endpoint.

> **Note** If you use a Linux operating system, you can run the following command to download the file:
>
> ```
> wget -c '<Public endpoint of the backup file, which is the download URL>' -O <File name>
> ```
>
> - The -c option enables resumable download.
> - The -O option saves the downloaded file by using the specified name. We recommend that you use the file name contained in the download URL.
> - If the URL contains more than one parameter, enclose the download URL in a pair of single quotation marks (').



9.1.12.5. Upload binlogs

Context

This topic describes how to upload binlog files to OSS.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.

3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration** to go to the **Backup and Restoration** page.
5. In the upper-right corner of the page, click **Upload Binlogs**.
6. In the message that appears, click **Confirm**.

9.1.12.6. Restore data to a new instance (formerly known as cloning an instance)

A cloned instance is a new instance with the same content as the primary instance, including data and settings. This feature allows you to restore data of the primary instance or create multiple instances that are the same as the primary instance.

Prerequisites

The following requirements are met:

- The primary instance is in the running state.
- The primary instance does not have an ongoing migration task.
- Data backup and log backup are enabled.
- The primary instance has at least one completed backup set before you clone the instance by backup set.

Context

You can specify a backup set or any point in time within the backup retention period to clone an instance.

> **Note**
> - A cloned instance copies only the content of the primary instance, but not the content of read-only instances. The copied content includes database information, account information, and instance settings such as whitelist settings, backup settings, parameter settings, and alert threshold settings.
> - The database engine of a cloned instance must be the same as that of the primary instance. Other settings can be different, such as the instance edition, zone, network type, instance type, and storage space. If you want to clone an instance to restore the data of a primary instance, we recommend that you select an instance type that has higher specifications and more storage space than those of the primary instance to speed up the data restoration process.
> - The account type of a cloned instance must be the same as that of the primary instance. The account password of the cloned instance can be changed.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. In the backup list, find a backup and click **Restore** in the Actions column.
6. In the dialog box that appears, select **Restore Database** and click **OK**.
7. On the **Restore Instance** page, configure the parameters described in the following table.

Section	Parameter	Description
Region	**Region**	The region where the ApsaraDB RDS instance resides.
Restore Database	**Restore Mode**	The data restoration mode of the primary instance. Valid values: ○ **By Time** ○ **By Backup Set**
	Restore Time	The point in time to which you want to restore the database. **Note** When **Restore Mode** is set to **By Time**, you must specify this parameter.
	Backup Set	The backup set for restoration. **Note** When **Restore Mode** is set to **By Backup Set**, you must specify this parameter.
Specifications	**Instance Name**	The name of the cloned instance.
	Database Engine	The engine of the database, which cannot be changed.
	Engine Version	The version of the database engine, which cannot be changed.
	Edition	The edition of the database. The actual values are displayed in the console.
	Storage Type	The storage type of the database. The actual values are displayed in the console.
	Instance Type	The type of the cloned instance. **Note** We recommend that you select an instance type and storage space that are higher than those of the primary instance to speed up the data restoration process.
	Storage Capacity	The storage space of the instance, including the space for data, system files, binlog files, and transaction files. The available storage capacity is displayed in the console. **Note** ApsaraDB RDS instances with local SSDs in the dedicated instance family occupy exclusive resources. The storage capacities are based on instance types.

Section	Parameter	Description
Network Type	Network Type	The network type of the instance. ApsaraDB RDS instances support the following network types: ○ **Classic Network**: Cloud services on the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ○ **VPC**: A VPC helps you build an isolated network environment on Apsara Stack. You can customize route tables, IP address ranges, and gateways within a VPC. We recommend that you select VPC for improved security.
	VPC	Select a VPC. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.
	vSwitch	Select a vSwitch. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.

8. After you configure the preceding parameters, click **Submit**.

9.1.13. CloudDBA

9.1.13.1. Introduction to CloudDBA

CloudDBA is a cloud service for database self-detection, self-repair, self-optimization, self-maintenance, and self-security ensuring based on machine learning and expert experience. CloudDBA helps you ensure stable, secure, and efficient databases without worrying about the management complexity and services failures caused by manual operations.

Features

In ApsaraDB RDS for MySQL, CloudDBA provides the following features:

- Diagnostics

  You can diagnose your instance and view the visualized diagnostic results.

- Instance sessions

  You can view sessions, collect session statistics, analyze SQL statements, and optimize the execution of SQL statements.

- Real-time monitoring

  You can view the real-time monitoring information of your instance, such as the queries per second (QPS), transactions per second (TPS), number of connections, and network traffic.

- Storage analysis

  You can view the space utilization, trends, exceptions, tablespaces, and data spaces.

- Deadlock analysis

You can view and analyze the last deadlock in a database.

- Dashboard

  You can view and compare performance trends, customize monitoring dashboards, check exceptions, and view instance topologies.

- Slow query logs

  You can view the trends and statistics of slow queries.

- Diagnostic reports

  You can use this feature to generate diagnostics reports or view automatically generated reports about instance health, alerts, and slow query logs.

9.1.13.2. Diagnostics

In ApsaraDB RDS for MySQL, CloudDBA provides the diagnostics feature. This feature diagnoses your ApsaraDB RDS for MySQL instance and visualizes the results.

Go to the Diagnostics page

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > Diagnostics**.
5. Click the **Diagnostics** tab.

> **Note** For more information, see Diagnostics in *Database Autonomy Service User Guide*.

9.1.13.3. Instance sessions

In ApsaraDB RDS for MySQL, CloudDBA provides the instance sessions feature. This feature allows you to view and manage sessions of an instance.

Open the Instance Sessions page

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > Diagnostics**.
5. Click the **Instance Sessions** tab.

> **Note** For more information, see Instance sessions in *DAS User Guide*.

9.1.13.4. Real-time monitoring

In ApsaraDB RDS for MySQL, CloudDBA provides the real-time monitoring feature. This feature allows you to view the real-time performance of your ApsaraDB RDS for MySQL instance.

Go to the Real-time Monitoring page

1. Log on to the ApsaraDB for RDS console.

2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > Diagnostics**.
5. Click the **Real-time Monitoring** tab.

> **Note** For more information, see Real-time monitoring in *Database Autonomy Service User Guide*.

9.1.13.5. Storage analysis

In ApsaraDB RDS for MySQL, CloudDBA provides the storage analysis feature. This feature allows you to check and solve storage exceptions in a timely manner to ensure database stability.

Context

You can use the storage analysis feature of CloudDBA to view the disk space usage of your ApsaraDB RDS for MySQL instance and the number of remaining days when disk space is available. It also provides information about the space usage, fragmentation, and exception diagnostic results of a table.

Go to the Storage Analysis page

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > Diagnostics**.
5. Click the **Storage Analysis** tab.

> **Note** For more information, see Storage analysis in *Database Autonomy Service User Guide*.

9.1.13.6. Deadlock analysis

In ApsaraDB RDS for MySQL, CloudDBA provides the deadlock analysis feature. This feature allows you to view and analyze the last deadlock in a database.

Open the Deadlock Analysis page

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > Diagnostics**.
5. Click the **Deadlock Analysis** tab.

> **Note** For more information, see Deadlock analysis in *DAS User Guide*.

9.1.13.7. Dashboard

In ApsaraDB RDS for MySQL, CloudDBA provides the dashboard feature. This feature allows you to view performance trends in specific ranges, compare performance trends, and customize charts to view performance trends.

Go to the Dashboard page

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > Dashboard**.

> **Note** For more information, see Performance trends in *Database Autonomy Service User Guide*.

9.1.13.8. Slow query logs

In ApsaraDB RDS for MySQL, CloudDBA provides the slow query logs feature. This feature allows you to view the trends and execution details of slow queries and obtain optimization suggestions for your ApsaraDB RDS for MySQL instance.

Go to the Slow Query Logs page

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > Slow Query Logs**.

> **Note** For more information, see Slow query logs in *Database Autonomy Service User Guide*.

9.1.13.9. Diagnostic reports

In ApsaraDB RDS for MySQL, CloudDBA provides the diagnostic reports feature. This feature allows you to create and view diagnostic reports.

Open the Report page

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > Diagnostic Reports**.

> **Note** For more information, see **View diagnostic reports** in *Database Autonomy Service User Guide*.

9.1.14. Logs

All ApsaraDB RDS instances support log management. You can query details about the error logs and slow query logs of an ApsaraDB RDS instance by using the ApsaraDB RDS console. The logs help you locate faults.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.

3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Logs**.
5. On the **Logs** page, click the **Error Logs**, **Slow Query Logs**, **Slow Query Log Summary**, or **Primary/Secondary Switching Logs** tab, select a time range, and then click **Search**.

Log type	Description
Error Logs	Records database running errors that occurred within the last month.
Slow Log Details	Records SQL statements within the last month that took longer than one second to execute. Duplicated SQL statements are removed. **Note** Slow query logs in the ApsaraDB RDS console are updated once every minute. However, you can query real-time slow query logs from the mysql.slow_log table.
Slow Query Log Summary	Records and analyzes SQL statements within the last month that took longer than one second to execute. Analysis reports of slow query logs are provided.
Primary/Secondary Switching Logs	Records the primary/secondary instance switching logs. This feature is applicable to ApsaraDB RDS for MySQL instances on High-availability Edition.

9.1.15. Use mysqldump to migrate MySQL data

This topic describes how to use mysqldump to migrate data from an on-premises database to an ApsaraDB RDS for MySQL instance.

Prerequisites

An ECS instance is created.

Context

mysqldump is easy to use but requires extensive downtime. This tool is suitable for scenarios where the amount of data is small or extensive downtime is allowed.

ApsaraDB RDS for MySQL is fully compatible with the native database service. The procedure of migrating data from the original database to an ApsaraDB RDS for MySQL instance is similar to that of migrating data from one MySQL server to another.

Before you migrate data, you must create an account that is used to migrate data from the on-premises MySQL database. You must grant the read and write permissions on the on-premises MySQL databases to the account.

Procedure

1. Run the following command to create a migration account for the on-premises database:

   ```
   CREATE USER 'username'@'host' IDENTIFIED BY 'password';
   ```

   Parameter description:

   - username: the name of the account to be created.
   - host: the host from which the account is authorized to log on to the on-premises MySQL database. If you want to allow access from a local host, set this parameter to localhost. If you want to allow access from all hosts, set this parameter to a percent sign (%).
   - password: the password of the account.

For example, you can run the following command to create an account with the username William and the password Changme123. The account is authorized to log on to the on-premises MySQL database from all hosts.

```
CREATE USER 'William'@'%' IDENTIFIED BY 'Changme123';
```

2. Run the following command to grant permissions to the migration account in the on-premises database:

```
GRANT SELECT ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION; GRANT REPLICATION SLAVE ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION;  GRANT REPLICATION SLAVE ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION;
```

Parameter description:

- privileges: the operation permissions granted to the account, such as SELECT, INSERT, and UPDATE. To authorize the account to perform all operations, set this parameter to ALL.
- databasename: the name of the on-premises MySQL database. If you want to grant all database permissions to the account, set this parameter to an asterisk (*).
- tablename: the name of the table whose data you want to migrate. If you want to grant all table permissions to the account, set this parameter to an asterisk (*).
- username: the name of the account.
- host: the host from which the account is authorized to log on to the on-premises MySQL database. If you want to allow access from a local host, set this parameter to localhost. If you want to allow access from all hosts, set this parameter to a percent sign (%).
- WITH GRANT OPTION: authorizes the account to use the GRANT statement. This parameter is optional.

For example, you can execute the following statement to grant all permissions on tables and databases to the William account. The account is authorized to log on to the database from all hosts.

```
GRANT ALL ON *. * TO 'William'@'%';
```

3. Use the data export tool of mysqldump to export data from the database as a data file.

> **Notice** Do not update data during data export. In this step, only data is exported. Stored procedures, triggers, and functions are not exported.

```
mysqldump -h localIp -u userName -p --opt --default-character-set=utf8 --hex-blob dbName --skip-triggers > /tmp/dbName.sql
```

Parameter description:

- localIp: the IP address of the host where the on-premises MySQL database resides.
- userName: the account that is used to migrate data from the on-premises MySQL database.
- dbName: the name of the on-premises MySQL database.
- /tmp/dbName.sql: the name of the exported data file.

4. Use mysqldump to export stored procedures, triggers, and functions.

> **Notice** Skip this step if no stored procedures, triggers, or functions are used in the database. When stored procedures, triggers, and functions are exported, you must remove the DEFINER to ensure compatibility with ApsaraDB RDS for MySQL.

```
mysqldump -h localIp -u userName -p --opt --default-character-set=utf8 --hex-blob dbName -R | sed -e 's/DEFINER[ ]*=[ ]*[^*]*\*/\*/' > /tmp/triggerProcedure.sql
```

Parameter description:

- localIp: the IP address of the host where the on-premises MySQL database resides.
- userName: the account that is used to migrate data from the on-premises MySQL database.
- dbName: the name of the on-premises MySQL database.

- /tmp/triggerProcedure.sql: the name of the exported stored procedure file.

5. Upload the data file and stored procedure file to the ECS instance.

   In this example, the files are uploaded to the following paths:

   ```
   /tmp/dbName.sql
   ```

   ```
   /tmp/triggerProcedure.sql
   ```

6. Log on to the ECS console and import both the data file and the stored procedure file to the destination ApsaraDB RDS for MySQL instance.

   > **Note** For information about how to log on to the ECS instance, see topics in the **Connect to an instance** section of ECS User Guide.

   ```
   mysql -h intranet4example.mysql.rds.aliyuncs.com -u userName -p dbName < /tmp/dbName.sql
   ```

   ```
   mysql -h intranet4example.mysql.rds.aliyuncs.com -u userName -p dbName < /tmp/triggerProcedure.sql
   ```

   Parameter description:

   - intranet4example.mysql.rds.aliyuncs.com: the endpoint of the ApsaraDB RDS for MySQL instance. An internal endpoint is used in this example.
   - userName: the migration account of the ApsaraDB RDS for MySQL database.
   - dbName: the name of the on-premises MySQL database from which you want to import data.
   - /tmp/dbName.sql: the name of the data file that you want to import.
   - /tmp/triggerProcedure.sql: the name of the stored procedure file that you want to import.

10.ApsaraDB RDS for SQL Server

10.1. User Guide(RDS SQL Server)

10.1.1. What is ApsaraDB RDS?

ApsaraDB RDS is a stable, reliable, and scalable online database service. Based on the distributed file system and high-performance storage, ApsaraDB RDS provides a set of solutions for disaster recovery, backup, restoration, monitoring, and migration.

ApsaraDB RDS supports four database engines, which are MySQL, SQL Server, PolarDB, and PostgreSQL. You can create database instances based on these engines to meet your business requirements. This topic describes the SQL Server engine.

ApsaraDB RDS for SQL Server

ApsaraDB RDS for SQL Server provides strong support for a variety of enterprise applications under the high-availability architecture. ApsaraDB RDS for SQL Server can also restore data to a specific point in time, which reduces costs.

ApsaraDB RDS for SQL Server provides basic features such as whitelist configuration, backup and restoration, transparent data encryption, data migration, and management for instances, accounts, and databases.

10.1.2. Log on to the ApsaraDB RDS console

This topic describes how to log on to the ApsaraDB RDS console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products** > Database Services > **ApsaraDB RDS**.

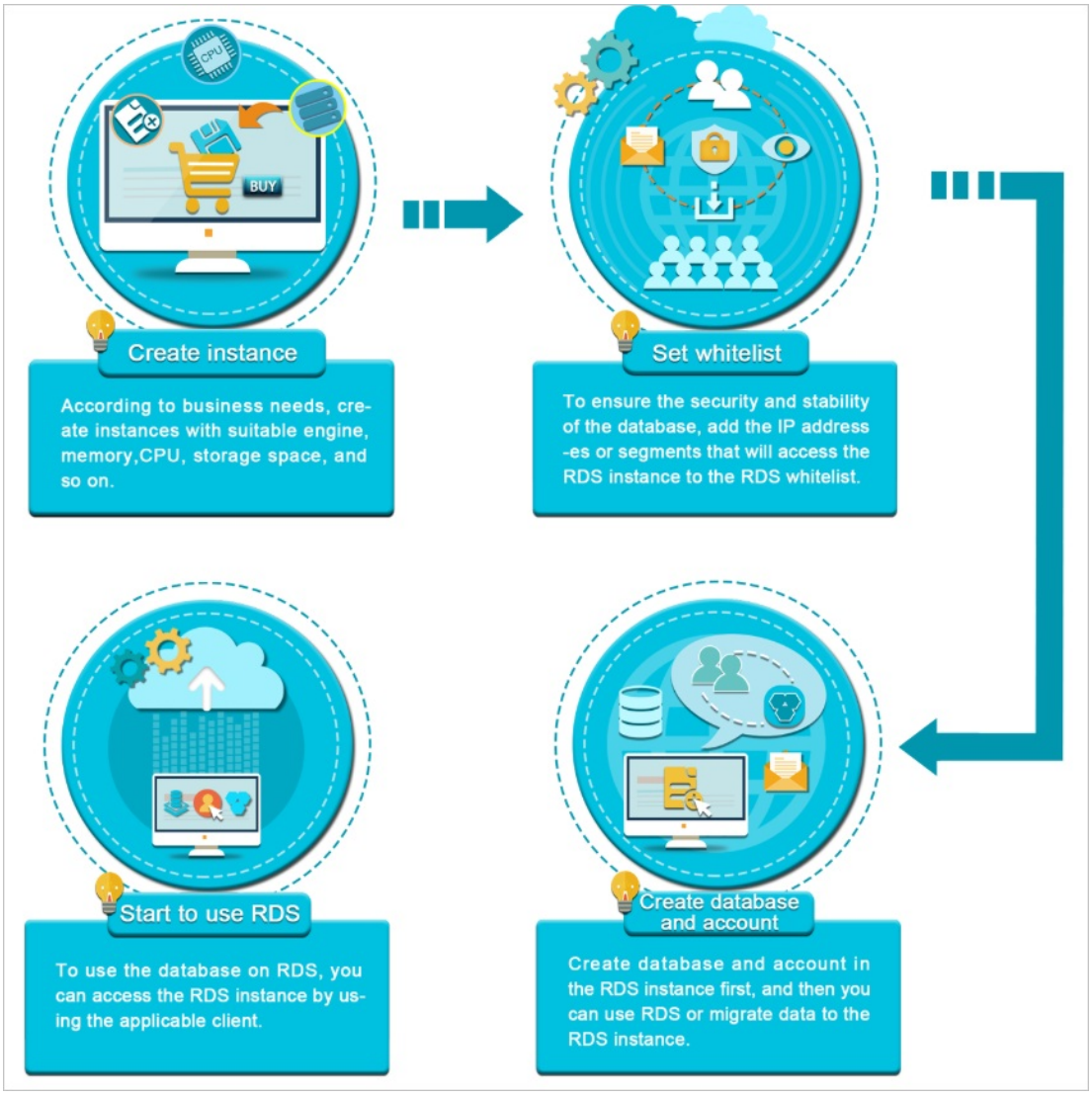
10.1.3. Quick Start

10.1.3.1. Procedure

ApsaraDB RDS quick start covers the following topics: creating an ApsaraDB RDS instance, configuring an IP address whitelist, creating a database, creating an account, and connecting to the instance.

The following figure shows the operations that you must perform before you use an ApsaraDB RDS instance.

Quick start flowchart



10.1.3.2. Create an instance

This topic describes how to create an instance in the ApsaraDB RDS console.

Prerequisites

An Apsara Stack account is created.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	Region	The region in which you want to create the instance. Services in different regions cannot communicate over an internal network. After the instance is created, the region cannot be changed.
	Zone of Primary Node	The zone where the primary instance is deployed.
	Deployment Method	Specifies whether to deploy the primary and secondary instances in separate zones. ApsaraDB RDS supports **Multi-zone Deployment** and **Single-zone Deployment**. If you select **Multi-zone Deployment**, you must configure **Zone of Secondary Node**.
	Zone of Secondary Node	The zone where the secondary instance is deployed. This parameter is available only when **Deployment Method** is set to **Multi-zone Deployment**. **Note** If you select the same zone for both the primary and secondary instances, the deployment is equivalent to single-zone deployment.
	Quantity	The number of ApsaraDB RDS instances that you want to create. Default value: 1.
	Instance Name	The name of the instance. ○ The name must be 2 to 64 characters in length. ○ The name must start with a letter. ○ The name can contain letters, digits, and the following special characters: _ - : ○ The name cannot start with http:// or https://.
	Network Type	The connection type of the instance. ApsaraDB RDS instances support the following connection types: ○ **Internet Connection**: ApsaraDB RDS instances of this connection type can be connected over the Internet. ○ **Internal Network**: ApsaraDB RDS instances of this connection type can be connected over an internal network. **Note** The value of this parameter cannot be changed after the instance is created. Proceed with caution.
	Database Engine	The database engine of the instance. Select **SQL Server**.

Section	Parameter	Description
Specifications	Engine Version	The version of the database engine. Valid values: ○ **2008r2**: SQL Server 2008 R2 ○ **2012_ent_ha**: SQL Server 2012 EE ○ **2012_std_ha**: SQL Server 2012 SE ○ **2016_ent_ha**: SQL Server 2016 EE ○ **2016_std_ha**: SQL Server 2016 SE ○ **2017_ent_ha**: SQL Server 2017 EE
	Edition	The edition of the instance. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Type	The storage type of the instance. This parameter is available only when you select an edition later than **2008r2**. Set the value to **Standard SSD**.
	Encrypted	Specifies whether to encrypt the standard SSD. This parameter is available only when **Storage Type** is set to **Standard SSD**. If you select Encrypted, you must specify the **Key** parameter. If you do not have a key, you must first create one in Key Management Service (KMS). For more information, see Create a CMK in *KMS User Guide*.
	Key	The key used to encrypt the standard SSD. This parameter is available only when you select **Encrypted**.
	Instance Type	The instance type of the instance. Memory size determines the maximum number of connections and the input/output operations per second (IOPS). The actual values are displayed in the console. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Capacity	The storage capacity of the instance, which includes the space to store data, system files, binlog files, and transaction files. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
Network	Network Type	The network type of the instance. ApsaraDB RDS instances support the following network types: ○ **Classic Network**: Cloud services in the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ○ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for improved security.
	VPC	Select a VPC. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.
	vSwitch	Select a vSwitch. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.

Section	Parameter	Description
	IP Address Whitelist	The IP addresses that are allowed to connect to the ApsaraDB RDS instance.

4. Click **Submit**.

10.1.3.3. Configure an IP address whitelist

After you create an ApsaraDB RDS instance, you must add the IP addresses or CIDR blocks that are used for database access to the IP address whitelist of the instance to ensure database security and reliability.

Context

IP address whitelists make your ApsaraDB RDS instance more secure and do not interrupt the operations of your ApsaraDB RDS instance when you configure whitelists. We recommend that you maintain your IP address whitelists on a regular basis.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Edit** corresponding to the **default** whitelist.

   > **Note** If you want to connect an ECS instance to an ApsaraDB RDS instance by using an internal endpoint, you must make sure that the two instances are in the same region and have the same network type. Otherwise, the connection fails.

6. In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks that are allowed to access the instance and click **OK**.

Note

- Limits for **IPv4** addresses:
  - You must separate multiple IP addresses with commas (,). A maximum of 1,000 different IP addresses can be added.

    Supported formats are `0.0.0.0/0`, IP addresses such as `10.23.12.24`, or CIDR blocks such as `10.23.12.24/24`. `/24` indicates the length of the IP address prefix in the CIDR block. An IP address prefix can contain 1 to 32 bits.
  - If an IP address whitelist is empty or contains `0.0.0.0/0`, all IP addresses can access the ApsaraDB RDS instance. This may cause security risks to the instance. Proceed with caution.
- Limits for **IPv6** addresses:
  - You must separate multiple IP addresses with commas (,). A maximum of 1,000 different IP addresses can be added.

    Supported formats are `::`, IP addresses such as `0:0:0:0:0:0:0:1`, or CIDR blocks such as `0:0:0:0:0:0:0:1/24`. `/24` indicates the length of the IP address prefix in the CIDR block. An IP address prefix can contain 1 to 128 bits.
  - If an IP address whitelist is empty or contains only `::`, all IP addresses can access the ApsaraDB RDS instance. This may cause security risks to the instance. Proceed with caution.
- You cannot specify both **IPv4** and **IPv6** addresses in a single IP address whitelist. If you want to specify both IPv4 and IPv6 addresses, specify them in separate IP address whitelists.
- If you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all the ECS instances that are created within your Apsara Stack tenant account appear. Then, you can select the IP addresses and add them to an IP address whitelist.

10.1.3.4. Connect to an instance

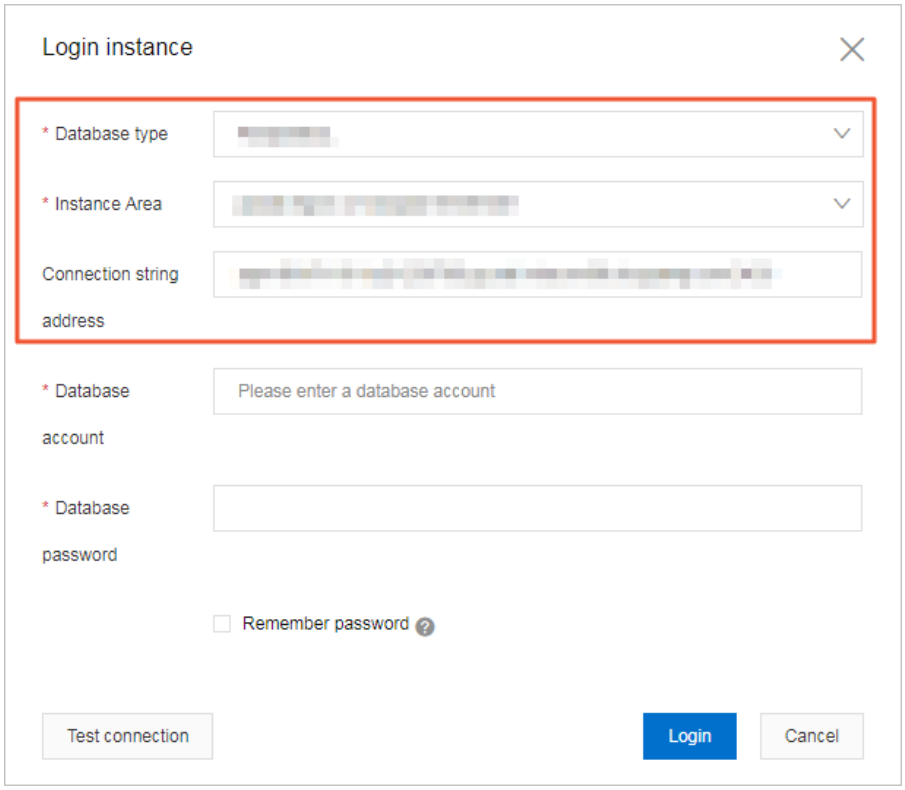
This topic describes how to use Data Management (DMS) to connect to an ApsaraDB RDS instance.

Prerequisites

- A dataset is created. For more information, see Create a database.
- A database account is created. For more information, see Create an account。

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database password**, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**.

> **Note**
> - If you want the browser to remember the password, select **Remember password** and click **Login**.
> - If you cannot connect to the instance, check the IP address whitelist settings. For more information, see Configure a whitelist.

10.1.3.5. Create an account on an ApsaraDB RDS instance that runs SQL Server 2017, 2016, or 2012

This topic describes how to create an account on an ApsaraDB RDS instance that runs SQL Server 2017, 2016, or 2012.

Prerequisites

- Your instance runs one of the following SQL Server versions and RDS editions:
  - SQL Server 2012 SE
  - SQL Server 2012 EE
  - SQL Server 2016 SE
  - SQL Server 2016 EE
  - SQL Server 2017 EE
- The instance is in the **Running** state.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. On the right side of the page, click **Create Account**.
6. Enter the information of the account to be created.

Parameter	Description
Database Account	Enter the name of the account. The name must be 2 to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter and end with a lowercase letter or digit.
Account Type	◦ **Privileged Account**: You can select the **Privileged Account** option only if you create an account on your ApsaraDB RDS instance for the first time. Each ApsaraDB RDS instance can have only a single privileged account. The privileged account of an ApsaraDB RDS instance cannot be deleted. ◦ **Standard Account**: You can select the **Standard Account** option only after a privileged account is created for your ApsaraDB RDS instance. Each ApsaraDB RDS instance can have more than one standard account. You must manually grant the permissions on databases to each standard account.

Parameter	Description
Authorized Databases	Select the authorized databases of the account when the **Standard Account** type is selected. If no databases are created, you can leave this parameter empty. You can perform the following steps to grant the permissions on more than one database to the account: i. In the **Unauthorized Databases** section, select the databases on which you want to grant permissions to the account. ii. Click the **>** icon to add the selected databases to the **Authorized Databases** section. iii. In the Authorized Databases section, specify the permissions that the account is granted on each authorized database. The permissions are **Read/Write**, **Read-only**, or **Owner**. You can also click **Set All to Read/Write**, **Set All to Read-only**, or **Set All to Owner** to set the permissions of the account on all authorized databases. **Note** ■ The account is authorized to create tables, delete tables, and modify schemas in a database only when it has the **Owner** permission on the database. ■ The account has permissions on all databases and does not require authorization if you have selected the **Privileged Account** type.
Password	Enter the password of the account. The password must meet the following requirements: ○ The password is 8 to 32 characters in length. ○ The password contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ Special characters include ! @ # $ % ^ & * ( ) _ + - =
Re-enter Password	Enter the password of the account again.
Description	Enter a description that helps identify the account. The description can be up to 256 characters in length.

7. Click **Create**.

Related information

- Create an account on an ApsaraDB RDS instance that runs SQL Server 2008 R2

10.1.3.6. Create a database

This topic describes how to create a database on an ApsaraDB RDS for SQL Server instance in the ApsaraDB RDS console.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.

4. In the left-side navigation pane, click **Databases**.
5. In the upper-right corner of the page, click **Create Database**.
6. Configure the parameters for the database that you want to create.

Parameter	Description
Database Name	Enter the name of the database. The name must be 2 to 64 characters in length. It can contain lowercase letters, digits, underscores (_), and hyphens (-). It must start with a lowercase letter and end with a lowercase letter or digit.
Supported Character Sets	Select the character set that is supported by the database. You can also select **all** and then select a character set from the drop-down list that appears.
Description	Enter a description of the database to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

10.1.4. Instances

10.1.4.1. Create an instance

This topic describes how to create an instance in the ApsaraDB RDS console.

Prerequisites

An Apsara Stack account is created.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	**Organization**	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	**Region**	The region in which you want to create the instance. Services in different regions cannot communicate over an internal network. After the instance is created, the region cannot be changed.
	Zone of Primary Node	The zone where the primary instance is deployed.
	Deployment Method	Specifies whether to deploy the primary and secondary instances in separate zones. ApsaraDB RDS supports **Multi-zone Deployment** and **Single-zone Deployment**. If you select **Multi-zone Deployment**, you must configure **Zone of Secondary Node**.

Section	Parameter	Description
	Zone of Secondary Node	The zone where the secondary instance is deployed. This parameter is available only when **Deployment Method** is set to **Multi-zone Deployment**. **Note** If you select the same zone for both the primary and secondary instances, the deployment is equivalent to single-zone deployment.
Specifications	**Quantity**	The number of ApsaraDB RDS instances that you want to create. Default value: 1.
	Instance Name	The name of the instance. ○ The name must be 2 to 64 characters in length. ○ The name must start with a letter. ○ The name can contain letters, digits, and the following special characters: _ - : ○ The name cannot start with http:// or https://.
	Network Type	The connection type of the instance. ApsaraDB RDS instances support the following connection types: ○ **Internet Connection**: ApsaraDB RDS instances of this connection type can be connected over the Internet. ○ **Internal Network**: ApsaraDB RDS instances of this connection type can be connected over an internal network. **Note** The value of this parameter cannot be changed after the instance is created. Proceed with caution.
	Database Engine	The database engine of the instance. Select **SQL Server**.
	Engine Version	The version of the database engine. Valid values: ○ **2008r2**: SQL Server 2008 R2 ○ **2012_ent_ha**: SQL Server 2012 EE ○ **2012_std_ha**: SQL Server 2012 SE ○ **2016_ent_ha**: SQL Server 2016 EE ○ **2016_std_ha**: SQL Server 2016 SE ○ **2017_ent_ha**: SQL Server 2017 EE
	Edition	The edition of the instance. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Type	The storage type of the instance. This parameter is available only when you select an edition later than **2008r2**. Set the value to **Standard SSD**.
	Encrypted	Specifies whether to encrypt the standard SSD. This parameter is available only when **Storage Type** is set to **Standard SSD**. If you select Encrypted, you must specify the **Key** parameter. If you do not have a key, you must first create one in Key Management Service (KMS). For more information, see Create a CMK in *KMS User Guide*.

Section	Parameter	Description
	Key	The key used to encrypt the standard SSD. This parameter is available only when you select **Encrypted**.
	Instance Type	The instance type of the instance. Memory size determines the maximum number of connections and the input/output operations per second (IOPS). The actual values are displayed in the console. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Capacity	The storage capacity of the instance, which includes the space to store data, system files, binlog files, and transaction files. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
Network	**Network Type**	The network type of the instance. ApsaraDB RDS instances support the following network types: ○ **Classic Network**: Cloud services in the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ○ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for improved security.
	VPC	Select a VPC. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.
	vSwitch	Select a vSwitch. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.
	IP Address Whitelist	The IP addresses that are allowed to connect to the ApsaraDB RDS instance.

4. Click **Submit**.

10.1.4.2. View basic information of an instance

This topic describes how to view the details of an ApsaraDB RDS instance, such as its basic information, internal network connection information, status, and configurations.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. Use one of the following methods to go to the **Basic Information** page of an instance:
   - On the **Instances** page, click the ID of an instance to go to the **Basic Information** page.
   - On the **Instances** page, click **Manage** in the **Actions** column corresponding to an instance to go to the **Basic Information** page.

10.1.4.3. Restart an instance

This topic describes how to manually restart an ApsaraDB RDS for MySQL instance. This applies if the number of connections exceeds a specific threshold or if an instance has performance issues.

Prerequisites

The instance is in the **Running** state.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click **Restart Instance** in the upper-right corner.

   > **Note** When you restart an instance, applications are disconnected from the instance. We recommend that you make appropriate service arrangements before you restart an instance. Proceed with caution.

4. In the Restart Instance message, click **Confirm**.

10.1.4.4. Change the specifications of an instance

This topic describes how to change specifications such as the instance type and storage space if they do not meet the requirements of your application. When the specification changes take effect, a 30-second network interruption may occur. Business operations that involve databases, accounts, and networks are interrupted. We recommend that you change the specifications during off-peak hours or make sure that your applications are configured with automatic reconnection policies.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section, click **Change Specifications**.
5. On the **Change Specifications** page, specify **Instance Type** and **Storage Capacity**.
6. After you configure the preceding parameters, click **Submit**.

10.1.4.5. Set a maintenance window

This topic describes how to set the maintenance window of an ApsaraDB RDS for SQL Server instance. The backend system performs maintenance on the ApsaraDB RDS instance during the maintenance window. This ensures the stability of the ApsaraDB RDS instance. The default maintenance window is from 02:00 (UTC+8) to 06:00 (UTC+8). We recommend that you set the maintenance window to off-peak hours of your business to avoid impacts on your business.

Context

- An instance enters the **Maintaining Instance** state before the maintenance window to ensure stability during the maintenance process. When the instance is in this state, access to data in the database and query operations such as performance monitoring are not affected. However, except for account and database management and IP address whitelist configuration, modification operations such as upgrade, downgrade, and restart are temporarily unavailable.
- During the maintenance window, one or two network interruptions may occur. Make sure that your applications are configured with automatic reconnection policies.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section, click **Configure** to the right of **Maintenance Window**.
5. Select a maintenance window and click **Save**.

> **Note** The maintenance window is displayed in UTC+8.

10.1.4.6. Configure primary/secondary switchover

ApsaraDB RDS provides the primary/secondary switchover feature to ensure the high availability of databases. The primary/secondary switchover is performed when the primary instance becomes unavailable. You can also manually switch your business to the secondary instance.

Prerequisites

The instance is in the **Running** state.

Context

An ApsaraDB RDS for SQL Server instance has a secondary instance. Data is synchronized in real time between the primary and secondary instances. You can access only the primary instance. The secondary instance serves only as a backup instance and does not allow external access. If the primary instance cannot be accessed, your business automatically switches over to the secondary instance. After the switchover, the primary instance becomes the secondary instance.

> **Notice**
>
> - During a switchover, a network interruption may occur. Make sure that your applications are configured with automatic reconnection policies.
> - During a switchover, a 1-minute data quality protection mechanism is enabled for data synchronization. If the primary and secondary database states are incorrect or if the latency for data synchronization exceeds 1 minute due to SQL Server errors, the HA system does not automatically perform the primary/secondary switchover. You must determine whether to perform the switchover.
> - If an instance is intermittently unavailable due to excessive mirroring event waits, the switchover is not performed. The instance automatically becomes available again.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Service Availability**.
5. In the **Availability Information** section, click **Switch Primary/Secondary Instance**.
6. In the dialog box that appears, click **OK**.

Result

After the switchover is complete, the original primary instance becomes the secondary instance for the next

primary/secondary switchover.

10.1.4.7. Release an instance

This topic describes how to manually release an instance.

Context

- Only instances in the running state can be manually released.
- After an instance is released, the instance data is immediately deleted. We recommend that you back up your data before you release an instance.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. Find the instance that you want to release and choose **More > Release Instance**.
3. In the **Release Instance** message, click **Confirm**.

10.1.4.8. Read-only instances

10.1.4.8.1. Overview of read-only ApsaraDB RDS for SQL Server instances

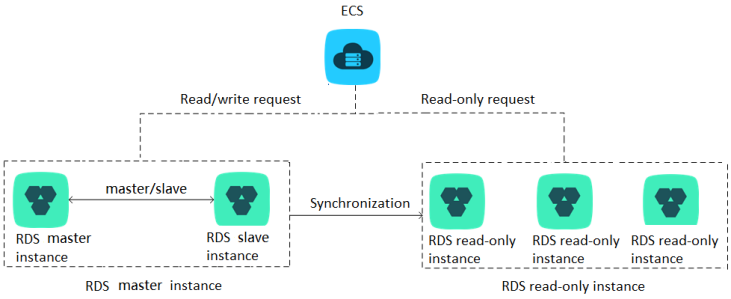
This topic provides an overview of read-only ApsaraDB RDS for SQL Server instances. If a large number of read requests overwhelm the primary instance, your business may be interrupted. In this case, you can create one or more read-only instances to offload read requests from the primary instance. This scales the read capability of your database system and increases the throughput of your application.

Overview

When a read-only instance is created, the data is replicated from the secondary instance. The data is consistent with that of the primary instance. Data updates of the primary instance are synchronized to all read-only instances.

> **Note**
> - Only ApsaraDB RDS instances that run SQL Server 2017 EE support read-only instances.
> - Each read-only instance works in a single-node architecture, where no instances are provided as backups.

The following figure shows the topology of read-only instances.



Features

- The specifications of a read-only instance can differ from the specifications of the primary instance, and can be

changed at any time. We recommend that you select specifications of a read-only instance that are higher than or equal to those of the primary instance. If the specifications of a read-only instance are lower than those of the primary instance, the read-only instance may have high latency or workloads.

- Read-only instances do not require database or account maintenance, because their database and account information is synchronized with the primary instance.
- A read-only instance automatically replicates the IP address whitelists of the primary instance. However, the IP address whitelists for the read-only instance are independent of those of the primary instance. For information about how to modify the whitelists of a read-only instance, see Configure a whitelist.
- You can monitor up to 20 system performance metrics, such as the disk capacity, input/output operations per second (IOPS), number of connections, CPU utilization, and network traffic.

Limits

- You can create up to seven read-only instances.
- You cannot configure backup policies or manually create backups for read-only instances, because these are already configured or created on the primary instance.
- You cannot create a temporary instance by using a backup set or from a point in time. In addition, you cannot overwrite a read-only instance by using a backup set.
- After a read-only instance is created, you cannot use a data backup file to restore it in overwrite mode.
- You cannot migrate data to read-only RDS instances.
- You cannot create or delete databases on read-only instances.
- You cannot create or delete accounts, authorize accounts, or change the passwords of accounts on read-only instances.

FAQ

Can I manage the accounts created on the primary instance from its read-only instances?

No, although accounts created on the primary instance are replicated to its read-only instances, you cannot manage the accounts on the read-only instances. The accounts have only read permissions on the read-only instances.

10.1.4.8.2. Create a read-only ApsaraDB RDS for SQL Server instance

This topic describes how to create a read-only instance for your primary ApsaraDB RDS for SQL server instance. This allows your database system to process a large number of read requests and increases the throughput of your application. Each read-only ApsaraDB RDS instance is a replica of the primary instance. Data updates on the primary instance are synchronized to all the read-only instances.

Prerequisites

The primary instance runs SQL Server 2017 EE.

Precautions

- You can create read-only instances for the primary ApsaraDB RDS instance. However, you cannot convert existing ApsaraDB RDS instances into read-only instances.
- While you create a read-only instance, the system replicates data from a secondary instance. Therefore, the operation of your primary instance is not interrupted.
- You can create up to seven read-only instances.
- For more information about read-only ApsaraDB RDS instances, see Overview of read-only ApsaraDB RDS for SQL Server instances.

Create a read-only instance

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the **Distributed by Instance Role** section on the right side of the page, click **Create Read-only Instance**.
5. Configure the following parameters and click **Submit**.

Section	Parameter	Description
Region	**Region**	The region in which you want to create the instance.
Specifications	**Database Engine**	The database engine of the read-only instance, which is the same as that of the primary instance and cannot be changed.
	Engine Version	The engine version of the read-only instance, which is the same as that of the primary instance and cannot be changed.
	Edition	Set the value to **Read-only**.
	Instance Type	The instance type of the read-only instance. The instance type of the read-only instance can be different from that of the primary instance, and can be changed at any time to facilitate flexible upgrade and downgrade. For more information, see Instance types in *ApsaraDB RDS Product Information*. **Note** To ensure sufficient I/O throughput for data synchronization, we recommend that you select at least the same instance type as the primary instance for read-only instances.
	Storage Capacity	The storage space of the read-only instance. To ensure sufficient I/O throughput for data synchronization, we recommend that you select at least the same instance type and storage space as the primary instance for the read-only instance. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
Network Type	**Network Type**	The network type of the read-only instance, which is the same as that of the primary instance and cannot be changed.
	VPC	Select a VPC if the network type is set to VPC.
	vSwitch	Select a vSwitch if the network type is set to VPC.

10.1.4.8.3. View details of read-only instances

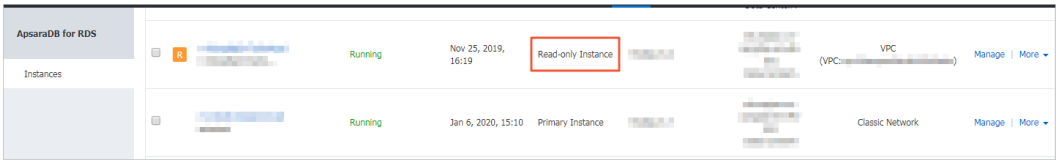
This topic describes how to view details of read-only instances. You can go to the Basic Information page of a read-only instance from the Instances page or the read-only instance list of the primary instance. Read-only instances are managed in the same way as primary instances. The read-only instance management page shows the management operations that can be performed.

View instance details by using a read-only instance

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, click the ID of a read-only instance. The **Basic Information** page appears.

   In the instance list, Instance Role of read-only instances is displayed as Read-only Instance, as shown in View a read-only instance.

   View a read-only instance



View instance details by using the primary instance

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. On the **Basic Information** page, move the pointer over the number below **Read-only Instance** in the **Distributed by Instance Role** section. The ID of the read-only instance is displayed.
5. Click the ID of the read-only instance to go to the read-only instance management page.

10.1.5. Accounts

10.1.5.1. Create an account on an ApsaraDB RDS instance that runs SQL Server 2017, 2016, or 2012

This topic describes how to create an account on an ApsaraDB RDS instance that runs SQL Server 2017, 2016, or 2012.

Prerequisites

- Your instance runs one of the following SQL Server versions and RDS editions:
  - SQL Server 2012 SE
  - SQL Server 2012 EE
  - SQL Server 2016 SE
  - SQL Server 2016 EE
  - SQL Server 2017 EE
- The instance is in the **Running** state.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic**

Information page.

4. In the left-side navigation pane, click **Accounts**.
5. On the right side of the page, click **Create Account**.
6. Enter the information of the account to be created.

Parameter	Description
Database Account	Enter the name of the account. The name must be 2 to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter and end with a lowercase letter or digit.
Account Type	○ **Privileged Account**: You can select the **Privileged Account** option only if you create an account on your ApsaraDB RDS instance for the first time. Each ApsaraDB RDS instance can have only a single privileged account. The privileged account of an ApsaraDB RDS instance cannot be deleted. ○ **Standard Account**: You can select the **Standard Account** option only after a privileged account is created for your ApsaraDB RDS instance. Each ApsaraDB RDS instance can have more than one standard account. You must manually grant the permissions on databases to each standard account.
Authorized Databases	Select the authorized databases of the account when the **Standard Account** type is selected. If no databases are created, you can leave this parameter empty. You can perform the following steps to grant the permissions on more than one database to the account: i. In the **Unauthorized Databases** section, select the databases on which you want to grant permissions to the account. ii. Click the **>** icon to add the selected databases to the **Authorized Databases** section. iii. In the Authorized Databases section, specify the permissions that the account is granted on each authorized database. The permissions are **Read/Write**, **Read-only**, or **Owner**. You can also click **Set All to Read/Write**, **Set All to Read-only**, or **Set All to Owner** to set the permissions of the account on all authorized databases. **Note** ■ The account is authorized to create tables, delete tables, and modify schemas in a database only when it has the **Owner** permission on the database. ■ The account has permissions on all databases and does not require authorization if you have selected the **Privileged Account** type.
Password	Enter the password of the account. The password must meet the following requirements: ○ The password is 8 to 32 characters in length. ○ The password contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ Special characters include ! @ # $ % ^ & * ( ) _ + - =
Re-enter Password	Enter the password of the account again.
Description	Enter a description that helps identify the account. The description can be up to 256 characters in length.

7. Click **Create**.

Related information

- Create an account on an ApsaraDB RDS instance that runs SQL Server 2008 R2

10.1.5.2. Reset the password

You can use the ApsaraDB RDS console to reset the password of your database account.

Prerequisites

The instance is in the **Running** state.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. In the left-side navigation pane, click **Accounts**.
4. Find an account and click **Reset Password** in the Actions column.
5. In the dialog box that appears, enter and confirm the new password, and then click **OK**.

> **Note** The password must meet the following requirements:
>
> - The password must be 8 to 32 characters in length.
> - The password must contain at least three of the following characters: uppercase letters, lowercase letters, digits, and special characters.
> - Special characters include ! @ # $ % ^ & * ( ) _ + - =

10.1.6. Databases

10.1.6.1. Create a database

This topic describes how to create a database on an ApsaraDB RDS for SQL Server instance.

Terms

- Instance: a virtualized database server on which you can create and manage more than one database.
- Database: a set of data that is stored in an organized manner and can be shared by a number of users. A database provides the minimal redundancy and is independent of applications. In simple words, a database is a data warehouse that is used to store data.
- Character set: a collection of letters, special characters, and encoding rules that are used in a database.

Prerequisites

An ApsaraDB RDS for SQL Server instance is created. For more information, see Create an instance.

Procedure

For more information, see Create a database.

10.1.6.2. Delete a database

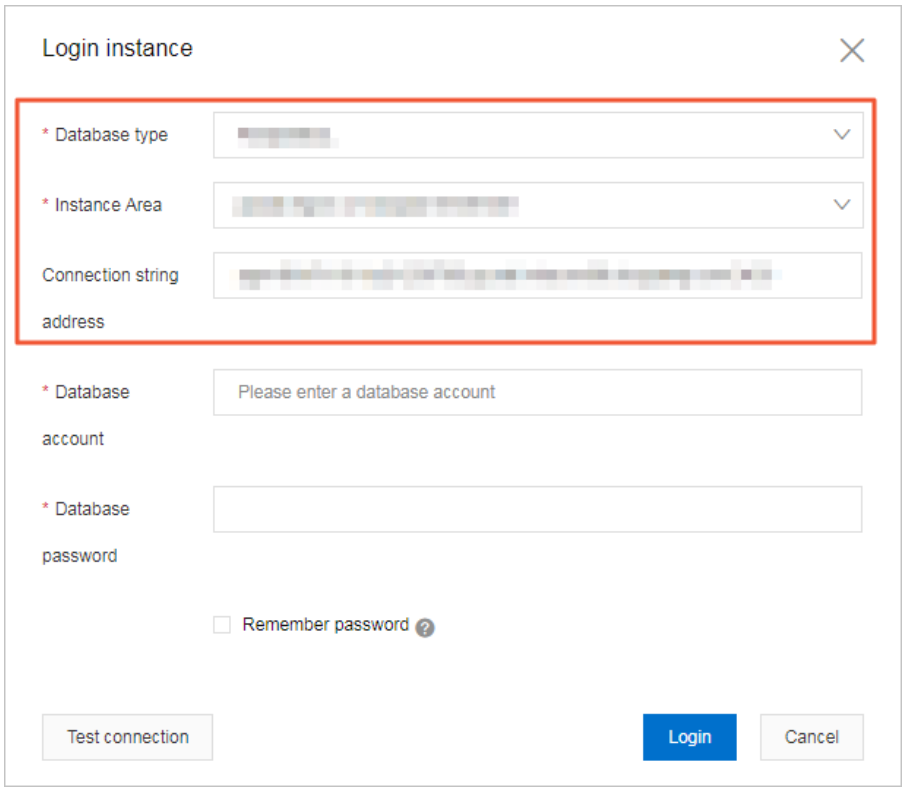
This topic describes how to delete a database from an ApsaraDB RDS for SQL Server instance. You can delete a database by using the ApsaraDB RDS console or an SQL statement.

Use the console to delete a database

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Databases**.
5. Find the database that you want to delete and click **Delete** in the **Actions** column.
6. In the message that appears, click **Confirm**.

Execute an SQL statement to delete a database

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database password**, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.

Parameter	Description
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**.

> **Note**
> - If you want the browser to remember the password, select **Remember password** and click **Login**.
> - If you cannot connect to the instance, check the IP address whitelist settings. For more information, see Configure a whitelist.

7. The **SQLConsole** page appears after you log on to the instance. Execute a statement in the following format to delete a database:

```
drop database <database name>;
```

> **Note** If the instance runs SQL Server 2012 or later on RDS High-availability Edition, you can also use the following stored procedure. This stored procedure deletes the specified database, removes the associated image, and closes the connection to the database.
>
> ```
> EXEC sp_rds_drop_database 'database name'
> ```

8. Click **execute**.

10.1.6.3. Change the character set collation and the time zone of system databases

This topic describes how to change the character set collation and the time zone of system databases. System databases include master, msdb, tempdb, and model.

Prerequisites

- The instance runs SQL Server 2012, 2016, or 2017.
- No database other than system databases exists on the instance.

> **Note** If you have just deleted databases from the instance, the deletion task may be pending in the secondary instance. Before you change the character set collation and the time zone, make sure that the primary and secondary instances do not contain databases.

Precautions

- The default character set collation is Chinese_PRC_CI_AS.
- The default time zone is China Standard Time.
- You can view the available character set collations and time zones in the console.
- The instance is in the unavailable state during the change process. It takes about 1 minute to change the time zone, and 2 to 10 minutes to change the character set collation.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Databases**.
5. On the Databases page, click **Change Character Set Collation and Time Zone**.

   > **Note** If you fail to find this button on the page, make sure that the requirements in Prerequisites are met.

6. In the dialog box that appears, select **Time Zone**, **Character Set Collation**, or both of them, and click **OK**.

UTC offsets of time zones

Time zone	UTC offset	Description
Afghanistan Standard Time	(UTC+04:30)	Kabul
Alaskan Standard Time	(UTC-09:00)	Alaska
Arabian Standard Time	(UTC+04:00)	Abu Dhabi, Muscat
Atlantic Standard Time	(UTC-04:00)	Atlantic Time (Canada)
AUS Central Standard Time	(UTC+09:30)	Darwin
AUS Eastern Standard Time	(UTC+10:00)	Canberra, Melbourne, Sydney
Belarus Standard Time	(UTC+03:00)	Minsk
Canada Central Standard Time	(UTC-06:00)	Saskatchewan
Cape Verde Standard Time	(UTC-01:00)	Cabo Verde Is.
Cen. Australia Standard Time	(UTC+09:30)	Adelaide
Central America Standard Time	(UTC-06:00)	Central America
Central Asia Standard Time	(UTC+06:00)	Astana
Central Brazilian Standard Time	(UTC-04:00)	Cuiaba
Central Europe Standard Time	(UTC+01:00)	Belgrade, Bratislava, Budapest, Ljubljana, Prague
Central European Standard Time	(UTC+01:00)	Sarajevo, Skopje, Warsaw, Zagreb
Central Pacific Standard Time	(UTC+11:00)	Solomon Islands, New Caledonia
Central Standard Time	(UTC-06:00)	Central Time (US and Canada)
Central Standard Time (Mexico)	(UTC-06:00)	Guadalajara, Mexico City, Monterrey
China Standard Time	(UTC+08:00)	Beijing, Chongqing, Hong Kong, Urumqi
E. Africa Standard Time	(UTC+03:00)	Nairobi

Time zone	UTC offset	Description
E. Australia Standard Time	(UTC+10:00)	Brisbane
E. Europe Standard Time	(UTC+02:00)	Chisinau
E. South America Standard Time	(UTC-03:00)	Brasilia
Eastern Standard Time	(UTC-05:00)	Eastern Time (US and Canada)
Georgian Standard Time	(UTC+04:00)	Tbilisi
GMT Standard Time	(UTC)	Dublin, Edinburgh, Lisbon, London
Greenland Standard Time	(UTC-03:00)	Greenland
Greenwich Standard Time	(UTC)	Monrovia, Reykjavik
GTB Standard Time	(UTC+02:00)	Athens, Bucharest
Hawaiian Standard Time	(UTC-10:00)	Hawaii
India Standard Time	(UTC+05:30)	Chennai, Kolkata, Mumbai, New Delhi
Jordan Standard Time	(UTC+02:00)	Amman
Korea Standard Time	(UTC+09:00)	Seoul
Middle East Standard Time	(UTC+02:00)	Beirut
Mountain Standard Time	(UTC-07:00)	Mountain Time (US and Canada)
Mountain Standard Time (Mexico)	(UTC-07:00)	Chihuahua, La Paz, Mazatlan
US Mountain Standard Time	(UTC-07:00)	Arizona
New Zealand Standard Time	(UTC+12:00)	Auckland, Wellington
Newfoundland Standard Time	(UTC-03:30)	Newfoundland
Pacific SA Standard Time	(UTC-03:00)	Santiago
Pacific Standard Time	(UTC-08:00)	Pacific Time (US and Canada)
Pacific Standard Time (Mexico)	(UTC-08:00)	Baja California
Russian Standard Time	(UTC+03:00)	Moscow, St. Petersburg, Volgograd
SA Pacific Standard Time	(UTC-05:00)	Bogota, Lima, Quito, Rio Branco
SE Asia Standard Time	(UTC+07:00)	Bangkok, Hanoi, Jakarta
China Standard Time	(UTC+08:00)	Kuala Lumpur, Singapore
Tokyo Standard Time	(UTC+09:00)	Osaka, Sapporo, Tokyo
US Eastern Standard Time	(UTC-05:00)	Indiana (East)
UTC	UTC	Coordinated Universal Time

Time zone	UTC offset	Description
UTC-02	(UTC-02:00)	Coordinated Universal Time-02
UTC-08	(UTC-08:00)	Coordinated Universal Time-08
UTC-09	(UTC-09:00)	Coordinated Universal Time-09
UTC-11	(UTC-11:00)	Coordinated Universal Time-11
UTC+12	(UTC+12:00)	Coordinated Universal Time+12
W. Australia Standard Time	(UTC+08:00)	Perth
W. Central Africa Standard Time	(UTC+01:00)	West Central Africa
W. Europe Standard Time	(UTC+01:00)	Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna

10.1.7. Database connection

10.1.7.1. Change the endpoint and port number of an instance

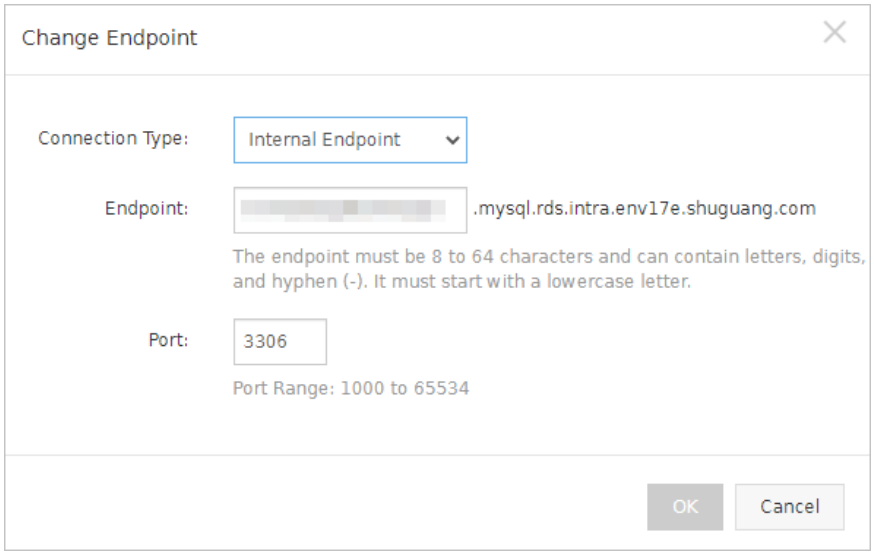
This topic describes how to view and change the endpoint and port number of an instance.

View the endpoint and port number

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
   You can view the internal endpoint and internal port of the instance in the **Database Connection** section.

Change the endpoint and port number

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the Database Connection section, click **Change Endpoint**.
6. In the dialog box that appears, set Connection Type, Endpoint, and Port, and click **OK**.



Note

- The prefix of the endpoint must be 8 to 64 characters in length and can contain only letters, digits, and hyphens (-). It must start with a lowercase letter.
- The port number must be in the range of 1000 to 5999.

10.1.7.2. Connect to an instance

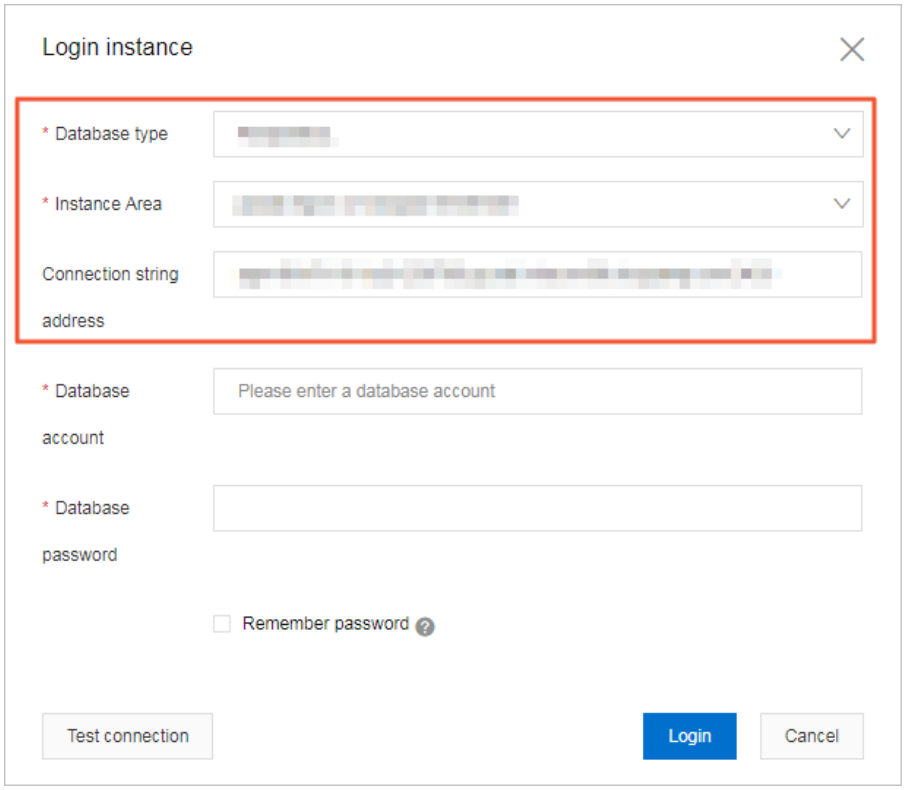
This topic describes how to use Data Management (DMS) to connect to an ApsaraDB RDS instance.

Prerequisites

- A dataset is created. For more information, see Create a database.
- A database account is created. For more information, see Create an account。

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database password**, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**.

> **Note**
> - If you want the browser to remember the password, select **Remember password** and click **Login**.
> - If you cannot connect to the instance, check the IP address whitelist settings. For more information, see Configure a whitelist.

10.1.8. Monitoring and alerting

10.1.8.1. Set a monitoring frequency

This topic describes how to set the monitoring frequency of an ApsaraDB RDS for SQL Server instance.

Context

ApsaraDB RDS provides the following monitoring frequencies:

- Every 60 seconds
- Every 300 seconds

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. On the **Resource Monitoring** tab, click **Set Monitoring Frequency**.
6. In the **Set Monitoring Frequency** dialog box, select the required monitoring frequency.
7. Click **OK**.

10.1.8.2. View resource and engine monitoring data

The ApsaraDB RDS console provides a variety of performance metrics to monitor the status of your instances.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. On the **Monitoring and Alerts** page, select **Resource Monitoring** or **Engine Monitoring**, and select a time range to view the corresponding monitoring data. The following table describes the metrics.

Monitoring type	Metric	Description
Resource Monitoring	Disk Space (unit: MB)	The disk usage of the instance, which includes the following items: ◦ Instance Size ◦ Data Usage ◦ Log Size ◦ Temporary File Size ◦ Other System File Size
	IOPS	The number of input/output operations per second (IOPS) for the instance.
	Total Connections	The total number of current connections of the instance.
	MSSQL Instance CPU Utilization (percentage in the operating system)	The CPU utilization of the instance. This includes the CPU utilization for the operating system. Unit: %.
	SQLServer Average Input/Output Traffic	The inbound and outbound traffic of the instance per second. Unit: KB.

Monitoring type	Metric	Description
Engine Monitoring	Average Transaction Frequency	The number of transactions processed per second.
	Average QPS	The number of SQL statements executed per second.
	Buffer Hit Ratio (%)	The read hit ratio of the buffer pool.
	Page Write Frequency at Check Point	The number of checkpoints written to pages per second.
	Login Frequency	The number of logons to the instance per second.
	Average Frequency of Whole Table Scans	The number of full table scans per second.
	SQL Compilations per Second	The number of SQL statements compiled per second.
	Lock Timeout Times	The number of lock timeouts on the instance per second.
	Deadlock Frequency	The number of deadlocks on the instance per second.
	Lock Wait Frequency	The number of lock waits on the instance per second.

10.1.9. Data security

10.1.9.1. Configure an IP address whitelist

After you create an ApsaraDB RDS instance, you must add the IP addresses or CIDR blocks that are used for database access to the IP address whitelist of the instance to ensure database security and reliability.

Context

IP address whitelists make your ApsaraDB RDS instance more secure and do not interrupt the operations of your ApsaraDB RDS instance when you configure whitelists. We recommend that you maintain your IP address whitelists on a regular basis.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Edit** corresponding to the **default** whitelist.

   > **Note** If you want to connect an ECS instance to an ApsaraDB RDS instance by using an internal endpoint, you must make sure that the two instances are in the same region and have the same network type. Otherwise, the connection fails.

6. In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks that are allowed to access the instance and click **OK**.

> **Note**
>
> - Limits for **IPv4** addresses:
>   - You must separate multiple IP addresses with commas (,). A maximum of 1,000 different IP addresses can be added.
>
>   Supported formats are `0.0.0.0/0`, IP addresses such as `10.23.12.24`, or CIDR blocks such as `10.23.12.24/24`. `/24` indicates the length of the IP address prefix in the CIDR block. An IP address prefix can contain 1 to 32 bits.
>   - If an IP address whitelist is empty or contains `0.0.0.0/0`, all IP addresses can access the ApsaraDB RDS instance. This may cause security risks to the instance. Proceed with caution.
> - Limits for **IPv6** addresses:
>   - You must separate multiple IP addresses with commas (,). A maximum of 1,000 different IP addresses can be added.
>
>   Supported formats are `::`, IP addresses such as `0:0:0:0:0:0:0:1`, or CIDR blocks such as `0:0:0:0:0:0:0:1/24`. `/24` indicates the length of the IP address prefix in the CIDR block. An IP address prefix can contain 1 to 128 bits.
>   - If an IP address whitelist is empty or contains only `::`, all IP addresses can access the ApsaraDB RDS instance. This may cause security risks to the instance. Proceed with caution.
> - You cannot specify both **IPv4** and **IPv6** addresses in a single IP address whitelist. If you want to specify both IPv4 and IPv6 addresses, specify them in separate IP address whitelists.
> - If you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all the ECS instances that are created within your Apsara Stack tenant account appear. Then, you can select the IP addresses and add them to an IP address whitelist.

10.1.9.2. Configure SSL encryption

This topic describes how to enhance endpoint security. You can enable Secure Sockets Layer (SSL) encryption and install SSL certificates that are issued by certificate authorities (CAs) to the required application services. SSL is used at the transport layer to encrypt network connections and enhance the security and integrity of communication data. However, SSL increases the response time.

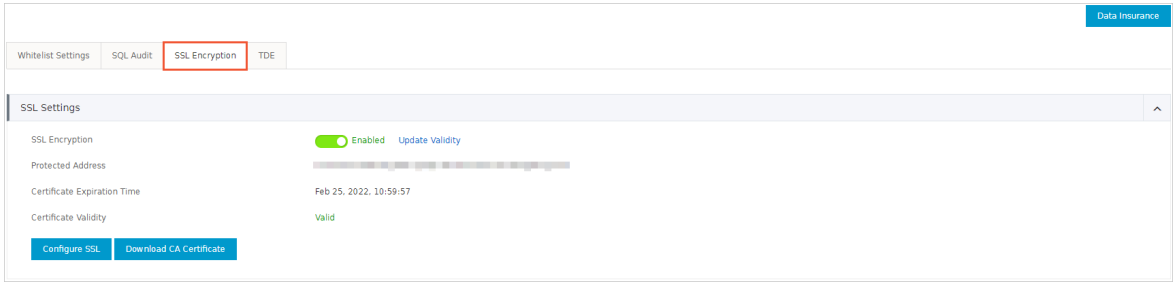
Precautions

- An SSL CA certificate is valid for one year. You must update the validity period of the SSL CA certificate in your application or client within one year. Otherwise, your application or client that uses encrypted network connections cannot connect to the ApsaraDB RDS instance.
- SSL encryption may cause a significant increase in CPU utilization. We recommend that you enable SSL encryption only when you want to encrypt connections from the Internet. In most cases, connections that use an internal endpoint do not require SSL encryption.
- SSL encryption cannot be disabled after it is enabled. Proceed with caution.

Enable SSL encryption

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SSL Encryption** tab.

6. In the SSL Settings section, turn on **SSL Encryption**.
7. In the **Configure SSL** dialog box, select the endpoint for which you want to enable SSL encryption and click **OK**.
8. Click **Download CA Certificate** to download the SSL CA certificate files in a compressed package.



The downloaded package contains three files:

- P7B file: used to import CA certificates to the Windows operating system.
- PEM file: used to import CA certificates to other operating systems or applications.
- JKS file: the Java truststore file. The password is apsaradb. It is used to import the CA certificate chain to Java programs.

> **Note** When the JKS file is used in Java, you must modify the default JDK security configuration in JDK 7 and JDK 8. Open the `/jre/lib/security/java.security` file on the host where your application resides, and modify the following configurations:
>
> ```
> jdk.tls.disabledAlgorithms=SSLv3, RC4, DH keySize < 224
> jdk.certpath.dis
> abledAlgorithms=MD2, RSA keySize < 1024
> ```
>
> If you do not modify the JDK security configuration, the following error is reported. Typically, other similar errors are also caused by invalid Java security configurations.
>
> ```
> javax.net.ssl.SSLHandshakeException: DHPublicKey does not comply to algorithm constraints
> ```

Configure an SSL CA certificate

After you enable SSL encryption, configure the SSL CA certificate on your application or client before they can connect to the ApsaraDB RDS instance. This section describes how to configure an SSL CA certificate. MySQL Workbench and Navicat are used in the example. For more information, see the instructions for the other applications or clients.

Configure a certificate on MySQL Workbench

1. Start MySQL Workbench.
2. Choose **Database > Manage Connections**.
3. Enable **Use SSL** and import the SSL CA certificate files.

Configure a certificate on Navicat

1. Start Navicat.
2. Right-click the database and select **Edit Connection**.
3. Click the **SSL** tab. Select the path of the PEM-formatted CA certificate, as shown in the following figure.
4. Click **OK**.

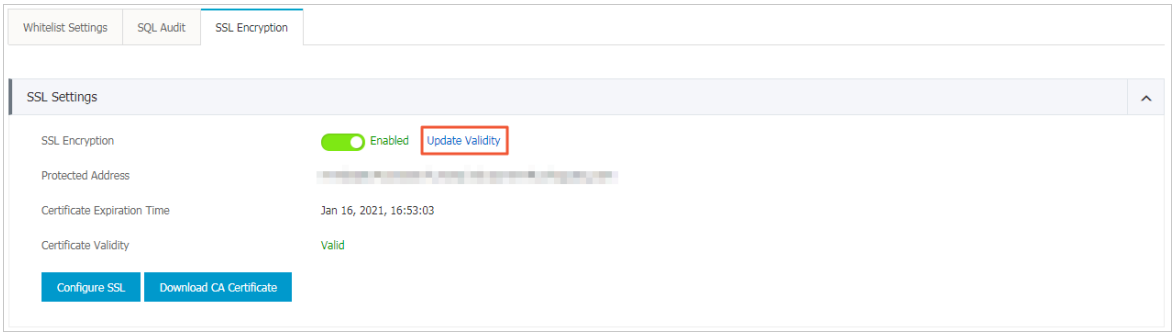
> **Note** If the `connection is being used` error is reported, the previous session is still connected. Restart Navicat.

5. Double-click the database to test whether the database is connected.

Update the validity period of an SSL CA certificate

> **Note**
> - **Update Validity** causes the ApsaraDB RDS instance to restart. Proceed with caution.
> - After you **update the validity period**, you must download and configure the SSL CA certificate again.

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SSL Encryption** tab.
6. Click **Update Validity**.



7. In the message that appears, click **OK**.

10.1.9.3. Configure TDE

This topic describes how to configure Transparent Data Encryption (TDE) for your ApsaraDB RDS for SQL Server instance. TDE allows your ApsaraDB RDS instance to encrypt the data that will be written into the disk and decrypt the data that will be read from the disk to the memory. TDE does not increase the sizes of data files. When you use TDE, you do not need to modify the application that uses the ApsaraDB RDS instance.

Precautions

- Instance-level TDE can be enabled but cannot be disabled. Database-level TDE can be enabled or disabled.
- The keys used for data encryption are generated and managed by Key Management Service (KMS). ApsaraDB RDS does not provide the keys or certificates used for data encryption. If you want to restore data to your computer after TDE is enabled, you must decrypt the data on your ApsaraDB RDS instance. For more information, see Decrypt data.
- TDE increases CPU utilization.

Prerequisites

- Your ApsaraDB RDS instance runs SQL Server EE.
- KMS is activated. If KMS is not activated, you can activate it as prompted when you enable TDE.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **TDE** tab.
6. Turn on **TDE Status**.
7. In the dialog box that appears, click **Confirm**.

   > **Note** If you have not enabled KMS, you are prompted to do so when you enable TDE. After you enable KMS, you can turn on **TDE Status** to enable TDE.

8. Click **Configure TDE**. In the Database TDE Settings dialog box, select the databases you want to encrypt from the Unselected Databases list, click the icon to add them to the **Selected Databases** list, and then click **OK**.

Decrypt data

If you want to decrypt a database that is encrypted by using TDE, you need only to remove the database from the Selected Databases section in the **Database TDE Settings** dialog box.

10.1.10. Database backup and restoration

10.1.10.1. Configure an automatic backup policy

Automatic backup supports full physical backups. ApsaraDB RDS automatically backs up data based on pre-configured policies. This topic describes how to configure a policy for automatic backup.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. On the **Backup and Restoration** page, click the **Backup Settings** tab.
6. Click **Edit**.
7. In the dialog box that appears, configure the automatic backup policy.

Parameter	Description
Data Retention Period	The number of days for which you want to retain data backup files. Valid values: 7 to 730. Default value: 7.

Parameter	Description
Backup Cycle	The cycle based on which you want to create a backup. You can select one or more days within a week. **Note** For data security purposes, we recommend that you back up your ApsaraDB RDS instance at least twice a week.
Backup Time	The period of time for which you want to back up data. Unit: hours.
Backup Frequency	The frequency at which you want to back up logs. The following options are available: ○ Same as Data Backup ○ Every 30 Minutes The total size of log backup files remains the same regardless of the backup frequency.

8. Click **OK**.

10.1.10.2. Manually back up an instance

This topic describes how to manually back up an ApsaraDB RDS instance.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. On the **Basic Information** page, click **Back Up Instance** in the upper-right corner.
5. In the **Back Up Instance** dialog box, select **Automatic Backup** or **Full Backup** from the **Select Backup Mode** drop-down list.

   Note ApsaraDB RDS supports the following backup methods:
   - **Automatic Backup**: After you select Automatic Backup, the system immediately performs an incremental or full backup based on the instance.
   - **Full Backup**: After you select Full Backup, the system immediately performs a full backup.

6. Click **OK**.

Result

After the backup is complete, you can view the backup task on the **Data Backup** tab of the **Backup and Restoration** page.

10.1.10.3. Shrink transaction logs

ApsaraDB RDS for SQL Server allows you to shrink transaction logs to reduce the log file size.

Prerequisites

The instance is in the **Running** state.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. In the upper-right corner of the page, click **Shrink Transaction Log**. In the message that appears, click **OK**.

> **Note** The shrinkage takes about 20 minutes to complete. ApsaraDB RDS for SQL Server shrinks transaction logs during each backup.

10.1.11. Migrate full backup data to ApsaraDB RDS for SQL Server

This topic describes how to migrate full backup files of an on-premises database from Object Storage Service (OSS) to ApsaraDB RDS for SQL Server.

Prerequisites

- Your ApsaraDB RDS instance has sufficient storage space. If the space is insufficient, you must increase it before you migrate data to the instance.
- The destination database on your ApsaraDB RDS instance has a different name from the on-premises database.
- A privileged account is created on your ApsaraDB RDS instance. For more information, see Create an account.
- An Object Storage Service (OSS) bucket is created in the region where your ApsaraDB RDS instance is created. For more information, see Create buckets in the *OSS User Guide*.
- The DBCC CHECKDB statement is executed, and the execution result indicates that no allocation or consistency errors occur.

> **Note** If no allocation or consistency errors occur, the following execution result is returned:

```
...
CHECKDB found 0 allocation errors and 0 consistency errors in database 'xxx'.
DBCC execution completed. If DBCC printed error messages, contact your system administrator.
```

Precautions

- Full backup files cannot be migrated to an ApsaraDB RDS instance of an earlier SQL Server version. For example, if the on-premises database runs SQL Server 2016 and your ApsaraDB RDS instance runs SQL Server 2012, you cannot migrate full backup files of the on-premises database to your ApsaraDB RDS instance.
- Differential or log backup files are not supported.
- The names of full backup files cannot contain special characters, such as `at signs (@) and vertical bars (|)`. If the file names contain special characters, the migration fails.
- After the service account of your ApsaraDB RDS instance is granted the access permission on the OSS bucket, the system creates a role named **AliyunRDSImportRole** in RAM. Do not modify or delete this role. Otherwise, you cannot download full backup files when you migrate data to your ApsaraDB RDS instance. In this case, you must re-authorize the service account of your ApsaraDB RDS instance.
- Before the migration is complete, do not delete the backup files from the OSS bucket. Otherwise, the migration fails.
- The names of backup files can be suffixed only with bak, diff, trn, or log. If you do not use the script in this

topic to generate a backup file, you must name the backup file by using one of the following suffixes:

- bak: indicates a full backup file.
- diff: indicates a differential backup file.
- trn or log: indicates a log backup file.

Back up the on-premises database

> **Note** Before you perform a full backup, stop writing data to the on-premises database. The data written during the backup process is not backed up.

1. Download the backup script. Double-click the backup script to open it by using the Microsoft SQL Server Management Studio (SSMS) client.
2. Configure the following parameters.

Parameter	Description
@backup_databases_list	The databases that you want to back up. Separate them with semicolons (;) or commas (,).
@backup_type	The backup type. Valid values: ○ FULL: full backup ○ DIFF: differential backup ○ LOG: log backup
@backup_folder	The directory in which you want to store the backup files on your computer. If the specified directory does not exist, the system creates a directory.
@is_run	Specifies whether to perform a backup. Valid values: ○ 1: performs a backup. ○ 0: performs no backup but a check.

3. Run the backup script.

Upload full backup files to the OSS bucket

After the on-premises database is backed up, you must upload full backup files to the OSS bucket. You can use one of the following methods:

- Use the OSS console

  If the size of backup files is smaller than 5 GB, you can upload the files in the OSS console. For more information, see Upload objects in the *OSS User Guide*.

- Call an OSS API operation

  You can call an OSS API operation to upload the full backup files in resumable mode. For more information, see Multipart upload-relevant operations in the *OSS Developer Guide*.

Create a migration task

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.

5. In the upper-right corner of the page, click **Migrate OSS Backup Data to RDS**.
6. Click **Next** twice until the Import Data step appears.
7. Configure the following parameters.

Parameter	Description
Database Name	Enter the name of the destination database on your ApsaraDB RDS instance. **Note** The name of the database must meet the requirements of SQL Server.
OSS Bucket	Select the OSS bucket that stores the backup files.
OSS Subfolder Name	Enter the name of the OSS subfolder that stores the backup files.
OSS File	Click the search icon to search for backup files by using the prefix-based fuzzy match. The system displays the name, size, and update time of each backup file. Select the backup file that you want to migrate to your ApsaraDB RDS instance.
Cloud Migration Method	**One-time Full Backup File Migration**: uploads full backup data to your ApsaraDB RDS instance. Select this option if you want to migrate only a single full backup file.

8. Click **OK**.

Wait for the migration task to complete. You can click **Refresh** to view the latest status of the migration task. If the migration fails, fix the error based on the message displayed in the Task Description column. For more information, see Common errors.

View the migration task

In the left-side navigation pane, click **Backup and Restoration**. Click the **Backup Data Upload History** tab. The system displays the migration tasks in the last week.

Common errors

Each record of a migration task contains a task description, which helps you identify the error cause and fix the error. The following list describes common errors:

- A database with the same name as the on-premises database exists on your ApsaraDB RDS instance.
  - Error message: The database (xxx) is already exist on RDS, please backup and drop it, then try again.
  - Cause: The on-premises database is named the same as an existing database on your ApsaraDB RDS instance. For data security purposes, ApsaraDB RDS for SQL Server does not allow such a database to be migrated.
  - Solution: If you need to overwrite the database in your ApsaraDB RDS instance with the on-premises database, you must back up the database, delete it from your ApsaraDB RDS instance, and then migrate the on-premises database to your ApsaraDB RDS instance.
- A differential backup file is used.
  - Error message: Backup set (xxx.bak) is a Database Differential backup, we only accept a FULL Backup.
  - Cause: The file that you uploaded is a differential backup file, but not a full backup file. The migration solution for full backup data supports only full backup files.
- A log backup file is used.
  - Error message: Backup set (xxx.trn) is a Transaction Log backup, we only accept a FULL Backup.
  - Cause: The file that you uploaded is a log backup file, but not a full backup file. The migration solution for full backup data supports only full backup files.

- The backup file fails the verification.
  - Error message: Failed to verify xxx.bak, backup file was corrupted or newer edition than RDS.
  - Cause: The backup file is damaged, or the on-premises database runs an SQL Server version later than your ApsaraDB RDS instance. For example, if the on-premises database runs SQL Server 2016 and your ApsaraDB RDS instance runs SQL Server 2012, the error message is returned.
  - Solution: If the backup file is damaged, perform a full backup on the on-premises database again. If the database engine version does not meet the requirements, select an ApsaraDB RDS instance that runs the same version as or a later version than the on-premises database.
- DBCC CHECKDB fails to be executed.
  - Error message: DBCC checkdb failed.
  - Cause: Allocation or consistency errors occurred in the on-premises database.
  - Solution: Execute the following statement in the on-premises database.

    Note Data loss may occur when you use this statement to fix errors.

    ```
    DBCC CHECKDB (DBName, REPAIR_ALLOW_DATA_LOSS) WITH NO_INFOMSGS, ALL_ERRORMSGS
    ```
- The remaining storage space of your ApsaraDB RDS instance is insufficient. (1)
  - Error message: Not Enough Disk Space for restoring, space left (xxx MB) < needed (xxx MB).
  - Cause: The remaining storage space of your ApsaraDB RDS instance does not meet the migration requirements.
  - Solution: Increase the storage space of your ApsaraDB RDS instance.
- The remaining storage space of your ApsaraDB RDS instance is insufficient. (2)
  - Error message: Not Enough Disk Space, space left xxx MB < bak file xxx MB.
  - Cause: The remaining storage space of your ApsaraDB RDS instance is smaller than the size of the backup file.
  - Solution: Increase the storage space of your ApsaraDB RDS instance.
- No privileged account exists.
  - Error message: Your RDS doesn't have any init account yet, please create one and grant permissions on RDS console to this migrated database (XXX).
  - Cause: No privileged account is created on your ApsaraDB RDS instance, and the database permissions are not granted to accounts. However, when this error message is returned, the backup file has been restored to your ApsaraDB RDS instance, and the migration task is successful.
  - Solution: Create a privileged account. For more information, see Create an account.

11.PolarDB

11.1. User Guide(PolarDB)

11.1.1. What is ApsaraDB RDS?

ApsaraDB RDS is a stable, reliable, and scalable online database service. Based on the distributed file system and high-performance storage, ApsaraDB RDS provides a set of solutions for disaster recovery, backup, restoration, monitoring, and migration.

ApsaraDB RDS supports four database engines, which are MySQL, SQL Server, PolarDB, and PostgreSQL. You can create database instances based on these engines to meet your business requirements.

PolarDB

PolarDB is a stable, secure, and scalable enterprise-grade relational database that provides one of the database engines that ApsaraDB RDS runs. Based on PostgreSQL, the most advanced open source database in the world, PolarDB enhances performance, application solutions, and compatibility. It also provides the capability of directly running Oracle applications. You can run enterprise-grade applications on PolarDB to implement stable and cost-effective services.

11.1.2. Limits on PolarDB

Before you use PolarDB, you must understand its limits and take the necessary precautions.

The following table describes the limits on PolarDB.

Operation	Limit
Database parameter modification	Not supported.
Root permissions of databases	Superuser permissions are not provided.
Database replication	• The system automatically builds HA databases based on PolarDB streaming replication without user input. • Secondary PolarDB instances are hidden and cannot be accessed.
Instance restart	Instances must be restarted by using the ApsaraDB RDS console or API operations.

11.1.3. Log on to the ApsaraDB RDS console

This topic describes how to log on to the ApsaraDB RDS console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password that you can use to log on to the console from the operations administrator.

> **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
>
> - Uppercase or lowercase letters
> - Digits
> - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products** > Database Services > **ApsaraDB RDS**.

11.1.4. Quick Start

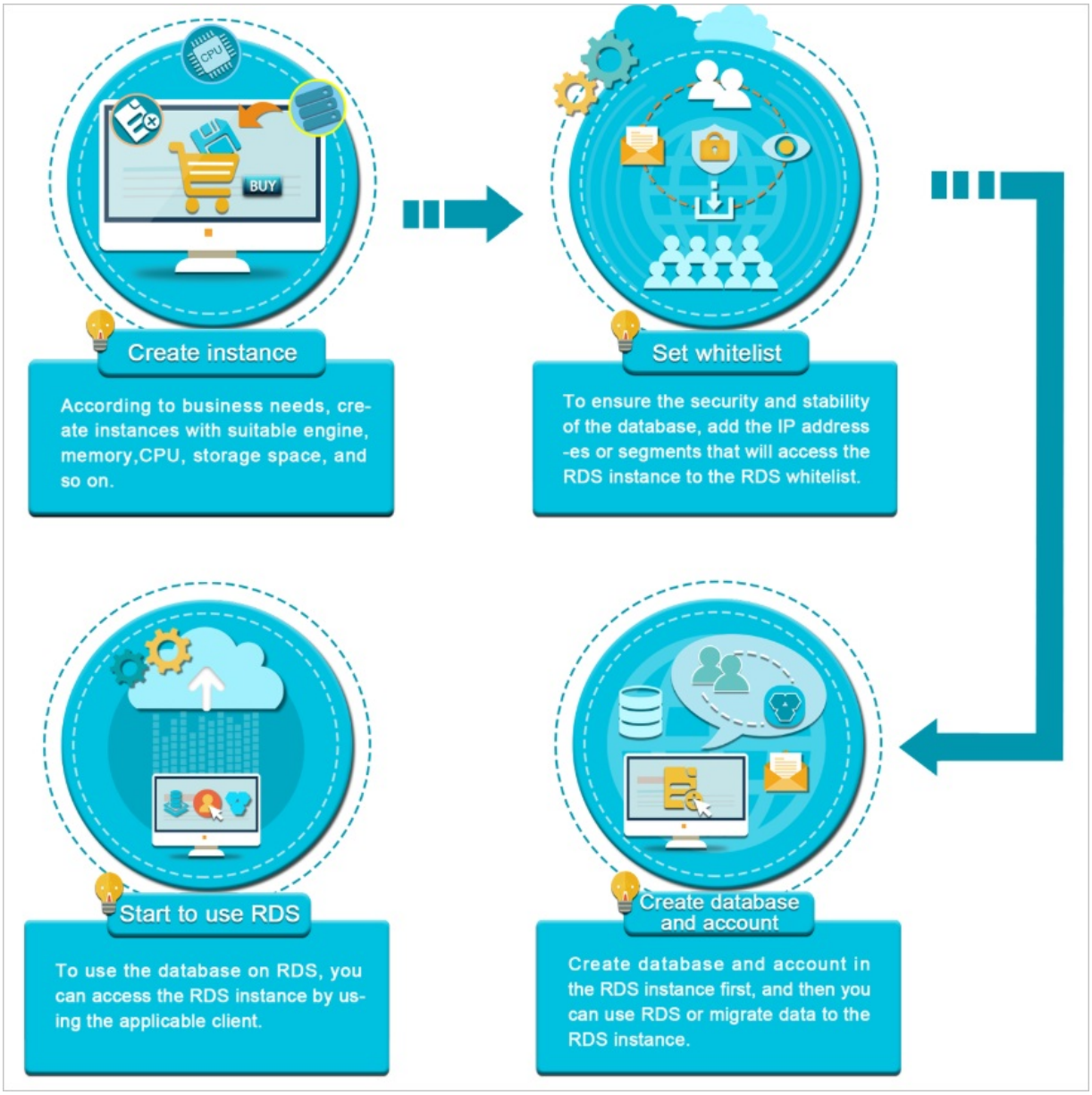
11.1.4.1. Procedure

ApsaraDB RDS quick start covers the following operations: creating an instance, configuring a whitelist, creating a database, creating an account, and connecting to the instance.

Flowchart for a PolarDB instance

If you are using ApsaraDB RDS for the first time, you can start with Limits.

The following figure shows the operations that you must perform before you use a PolarDB instance.



11.1.4.2. Create an instance

This topic describes how to create a PolarDB instance in the console.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	**Organization**	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.

Section	Parameter	Description
Region	**Region**	The region in which you want to create the instance. Services in different regions cannot communicate over an internal network. After the instance is created, the region cannot be changed.
	Zone of Primary Node	The zone where the primary instance is deployed.
	Deployment Method	Specifies whether to deploy the primary and secondary instances in separate zones. ApsaraDB RDS supports **Multi-zone Deployment** and **Single-zone Deployment**. If you select **Multi-zone Deployment**, you must configure **Zone of Secondary Node**.
	Zone of Secondary Node	The zone where the secondary instance is deployed. This parameter is available only when **Deployment Method** is set to **Multi-zone Deployment**. **Note** If you select the same zone for both the primary and secondary instances, the deployment is equivalent to single-zone deployment.
Specifications	**Quantity**	The number of ApsaraDB RDS instances that you want to create. Default value: 1.
	Instance Name	The name of the instance. ○ The name must be 2 to 64 characters in length. ○ The name must start with a letter. ○ The name can contain letters, digits, and the following special characters: _-: ○ The name cannot start with http:// or https://.
	Network Type	The connection type of the instance. ApsaraDB RDS instances support the following connection types: ○ **Internet Connection**: ApsaraDB RDS instances of this connection type can be connected over the Internet. ○ **Internal Network**: ApsaraDB RDS instances of this connection type can be connected over an internal network. **Note** The value of this parameter cannot be changed after the instance is created. Proceed with caution.
	Database Engine	The database engine of the instance. Select **POLARDB**.
	Engine Version	The version of the database engine. Valid values: 11.
	Edition	The edition of the instance. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Type	The storage type of the instance. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.

Section	Parameter	Description
	Instance Type	The instance type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Capacity	The storage capacity of the instance, which includes the space to store data, system files, binlog files, and transaction files. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Network	**Network Type**
		VPC
vSwitch		The vSwitch in the VPC. **Note** You must specify this parameter when **Network Type** is set to **VPC**.
IP Address Whitelist		The IP addresses that are allowed to connect to the ApsaraDB RDS instance.

4. After you configure the preceding parameters, click **Submit**.

11.1.4.3. Configure an IP address whitelist

This topic describes how to configure a whitelist for a PolarDB instance. Only entities that are listed in a whitelist can access your PolarDB instance.

Context

Whitelists make your PolarDB instance more secure and do not interrupt the operations of your PolarDB instance when you configure whitelists. We recommend that you perform maintenance on your whitelists on a regular basis.

To configure a whitelist, perform the following operations:

- Configure a whitelist: Add IP addresses to allow them to connect to the PolarDB instance.

  Note The default IP address whitelist contains only the IP address 127.0.0.1. This indicates that no devices are allowed to access the PolarDB instance.

- Configure an ECS security group: Add an ECS security group for the PolarDB instance to allow ECS instances in the group to connect to the PolarDB instance.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Edit** corresponding to the **default** whitelist.

   Note You can also click **Create Whitelist** to create a whitelist.

6. In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks that are allowed to access the instance and click **OK**. The following section describes the rules:
   - If you enter the CIDR block 10.10.10.0/24 in the IP Addresses field, all IP addresses in the 10.10.10.X format can access your PolarDB instance.
   - If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
   - If you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all ECS instances created within your Apsara Stack account are displayed. You can select the required IP addresses to add them to the IP address whitelist.

11.1.4.4. Create a database and an account

Before you start to use PolarDB, you must create a database and an account for a PolarDB instance. This topic describes how to create a database and an account for a PolarDB instance.

Create an account

You can create privileged and standard accounts on a PolarDB instance. The following section describes how to create a privileged account.

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Create Privileged Account** and configure the following parameters.

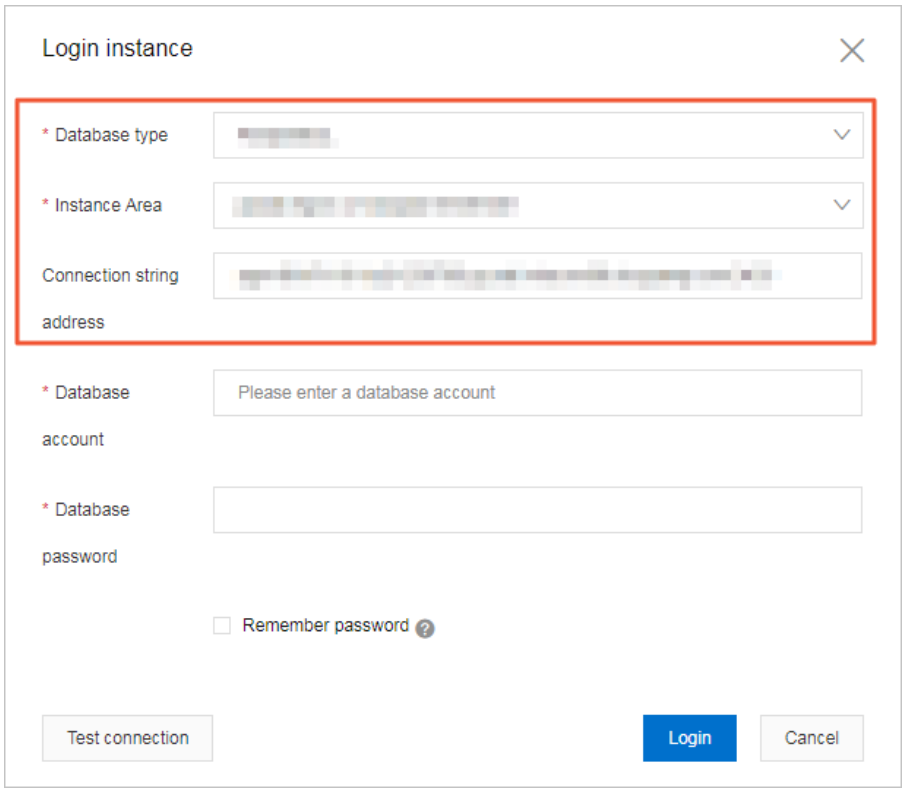
Parameter	Description
Database Account	○ The account name must be 2 to 16 characters in length. ○ The account name can contain lowercase letters, digits, and underscores (_). ○ The account name must start with a lowercase letter and end with a lowercase letter or digit.
Password	○ The password of the account must be 8 to 32 characters in length. ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ Special characters include ! @ # $ % ^ & * ( ) _ + - =
Re-enter Password	Enter the password of the account again.

6. Click **Create**.

Create a database and a standard account

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database**

password, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**. If you want the browser to remember the password, select **Remember password** before you click **Login**.

   > **Note** If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see Configure an IP whitelist.

7. The **SQLConsole** page appears after you log on to the instance. Execute a statement in the following format to create a database:

```
CREATE DATABASE name
 [ [ WITH ] [ OWNER [=] user_name ]
        [ TEMPLATE [=] template ]
        [ ENCODING [=] encoding ]
        [ LC_COLLATE [=] lc_collate ]
        [ LC_CTYPE [=] lc_ctype ]
        [ TABLESPACE [=] tablespace_name ]
        [ CONNECTION LIMIT [=] connlimit ] ]
```

For example, if you want to create a database named test, execute the following statement:

```
create database test;
```

8. Click **execute**.
9. In the SQL window, execute a statement in the following format to create a standard account:

```
CREATE USER name [ [ WITH ] option [ ... ] ]
where option can be:
   SUPERUSER | NOSUPERUSER
 | CREATEDB | NOCREATEDB
 | CREATEROLE | NOCREATEROLE
 | CREATEUSER | NOCREATEUSER
 | INHERIT | NOINHERIT
 | LOGIN | NOLOGIN
 | REPLICATION | NOREPLICATION
 | CONNECTION LIMIT connlimit
 | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
 | VALID UNTIL 'timestamp'
 | IN ROLE role_name [, ...]
 | IN GROUP role_name [, ...]
 | ROLE role_name [, ...]
 | ADMIN role_name [, ...]
 | USER role_name [, ...]
 | SYSID uid
```

For example, if you want to create a user account named test2 whose password is 123456, execute the following statement:

```
create user test2 password '123456';
```

10. Click **execute**.

11.1.4.5. Connect to an instance

This topic describes how to use Data Management (DMS) or the pgAdmin 4 client to connect to an ApsaraDB RDS instance.

Context

You can log on to DMS from the ApsaraDB RDS console and then connect to an ApsaraDB RDS instance.

Data Management (DMS) is a data management service that integrates data, schema, and server management, access security, BI charts, data trends, data tracking, and performance optimization. DMS can be used to manage relational and non-relational databases, such as MySQL, SQL Server, and PostgreSQL. It can also be used to manage Linux servers.

You can also use a database client to connect to an ApsaraDB RDS instance. ApsaraDB RDS is fully compatible with PolarDB. You can connect to PolarDB instances in the similar manner as you would connect to an open source PolarDB database. In this topic, the pgAdmin 4 client is used to connect to an ApsaraDB RDS instance.

Use DMS to connect to an ApsaraDB RDS instance

For more information about how to use DMS to connect to an ApsaraDB RDS instance, see Log on to an ApsaraDB for RDS instance by using DMS.

Use the pgAdmin 4 client to connect to an ApsaraDB RDS instance

1. Add the IP address of the pgAdmin client to an IP address whitelist of the ApsaraDB RDS instance. For more information about how to configure a whitelist, see Configure an IP whitelist.
2. Start the pgAdmin 4 client.

   > **Note** For information about how to download the pgAdmin 4 client, visit pgAdmin 4 (Windows).

3. Right-click **Servers** and choose **Create > Server**, as shown in the following figure.
4. On the **General** tab of the **Create - Server** dialog box, enter the name of the server, as shown in the following figure.
5. Click the **Connection** tab and enter the information of the instance, as shown in the following figure.

Parameter	Description
Host name/address	The internal endpoint of the ApsaraDB RDS instance. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.
Port	The internal port number that is used to connect to the ApsaraDB RDS instance. For more information about how to view the internal port number, see View and modify the internal endpoint and port number.
Username	The name of the privileged account for the ApsaraDB RDS instance. For more information about how to create an privileged account, see Create a database and an account.
Password	The password of the privileged account.

6. Click **Save**.
7. If the connection information is correct, choose **Servers > Server Name > Databases > postgres**. If the following page appears, the connection is established.

   > **Notice** The postgres database is the default system database of the ApsaraDB RDS instance. Do not perform operations on this database.

11.1.5. Instances

11.1.5.1. Create an instance

This topic describes how to create a PolarDB instance in the console.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	Region	The region in which you want to create the instance. Services in different regions cannot communicate over an internal network. After the instance is created, the region cannot be changed.
	Zone of Primary Node	The zone where the primary instance is deployed.
	Deployment Method	Specifies whether to deploy the primary and secondary instances in separate zones. ApsaraDB RDS supports **Multi-zone Deployment** and **Single-zone Deployment**. If you select **Multi-zone Deployment**, you must configure **Zone of Secondary Node**.
	Zone of Secondary Node	The zone where the secondary instance is deployed. This parameter is available only when **Deployment Method** is set to **Multi-zone Deployment**. **Note** If you select the same zone for both the primary and secondary instances, the deployment is equivalent to single-zone deployment.
Specifications	Quantity	The number of ApsaraDB RDS instances that you want to create. Default value: 1.
	Instance Name	The name of the instance. ○ The name must be 2 to 64 characters in length. ○ The name must start with a letter. ○ The name can contain letters, digits, and the following special characters: _-: ○ The name cannot start with http:// or https://.
	Network Type	The connection type of the instance. ApsaraDB RDS instances support the following connection types: ○ **Internet Connection**: ApsaraDB RDS instances of this connection type can be connected over the Internet. ○ **Internal Network**: ApsaraDB RDS instances of this connection type can be connected over an internal network. **Note** The value of this parameter cannot be changed after the instance is created. Proceed with caution.
	Database Engine	The database engine of the instance. Select **POLARDB**.
	Engine Version	The version of the database engine. Valid values: 11.
	Edition	The edition of the instance. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.

Section	Parameter	Description
	Storage Type	The storage type of the instance. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Instance Type	The instance type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Capacity	The storage capacity of the instance, which includes the space to store data, system files, binlog files, and transaction files. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Network	**Network Type**
		VPC
vSwitch		The vSwitch in the VPC. **Note** You must specify this parameter when **Network Type** is set to **VPC**.
IP Address Whitelist		The IP addresses that are allowed to connect to the ApsaraDB RDS instance.

4. After you configure the preceding parameters, click **Submit**.

11.1.5.2. Restart an instance

This topic describes how to manually restart an ApsaraDB RDS instance. This applies if the number of connections exceeds a specific threshold or if an instance has performance issues.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Restart Instance**.

   Notice When you restart an instance, a network interruption occurs. We recommend that you make appropriate arrangements for your workloads and make sure that your applications are configured with automatic reconnection policies.

5. In the message that appears, click **Confirm**.

11.1.5.3. Set a maintenance window

This topic describes how to set a maintenance window for an ApsaraDB RDS instance. To ensure the stability of ApsaraDB RDS instances, the backend system performs maintenance of the instances at irregular intervals. The default maintenance window is from 02:00 to 06:00. You can set the maintenance window to the off-peak period of your business to avoid impact on business.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section, click **Configure** to the right of **Maintenance Window**.
5. Select a maintenance window and click **Save**.

> **Note**
> - The maintenance window is in UTC+8.
> - Before the maintenance starts, ApsaraDB RDS sends text messages and emails to the contacts that are associated with your Apsara Stack account.
> - To ensure the stability of the maintenance process, the instance changes to the Maintaining Instance state before the maintenance window. When the instance is in this state, access to data in the database and query operations such as performance monitoring are not affected. However, apart from account and database management and IP address whitelist configuration, modification operations such as upgrade, downgrade, and restart are temporarily unavailable.
> - During the maintenance window, one or two network interruptions may occur. Make sure that you configure automatic reconnection policies for your applications to avoid service disruptions.

11.1.5.4. Configure primary/secondary switchover

ApsaraDB RDS provides the primary/secondary switchover feature to ensure the high availability of databases. The primary/secondary switchover is performed when the primary instance becomes unavailable. You can also manually switch your business to the secondary instance. This topic describes how to manually switch over services between a primary instance and its secondary instance.

Context

Data is synchronized in real time between the primary and secondary instances. You can access only the primary instance. The secondary instance serves only as a backup instance and does not allow external access. After the switchover, the original primary instance becomes the secondary instance.

> **Note** During a switchover, a network interruption may occur. Make sure that your applications are configured with automatic reconnection policies.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Service Availability**.
5. In the **Availability Information** section, click **Switch Primary/Secondary Instance**.
6. In the **Switch Primary/Secondary Instance** dialog box, click **OK**.

> Note
> - During the switchover, operations such as managing databases and accounts and changing network types cannot be performed. Therefore, we recommend that you select Switch Within Maintenance Window.
> - For more information about how to set a maintenance window, see Set the maintenance window.

11.1.5.5. Release an instance

This topic describes how to manually release a PolarDB instance.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. In the **Actions** column corresponding to the instance, click **More** and select **Release Instance** from the drop-down list.

   > Note After an instance is released, the instance data is immediately deleted. We recommend that you back up the data and download the backup file before you release an instance. For more information, see Back up data and Download backup files.

4. In the message that appears, click **Confirm**.

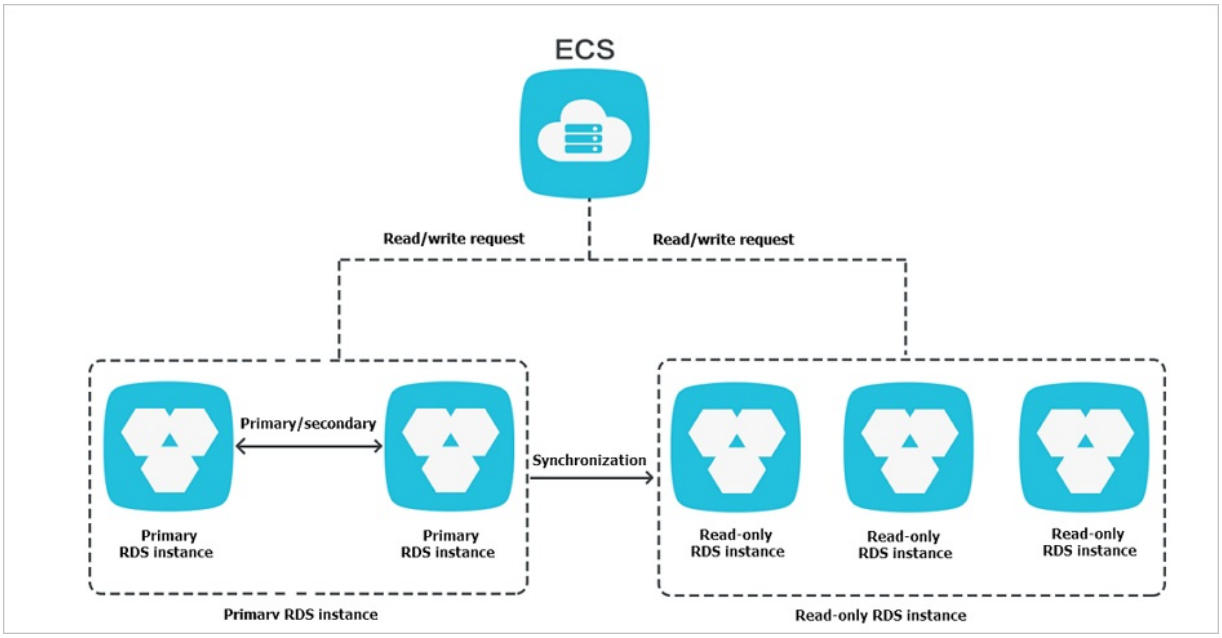
11.1.5.6. Read-only instance

11.1.5.6.1. Overview

PolarDB allows you to create read-only instances. In scenarios where PolarDB has a small number of write requests but a large number of read requests, you can create read-only instances to distribute database access loads away from the primary instance. This topic describes the features and limits of read-only instances.

If a large number of read requests overwhelm the primary instance, your business may be interrupted. In this case, you can create one or more read-only instances to offload read requests from the primary instance. This scales the read capability of your database system and increases the throughput of your application.

Each read-only instance works in a single-node architecture, where no instances are provided as backups. Data updates of the primary instance are automatically synchronized to all read-only instances immediately after the primary instance completes operations. Read-only instances reside within the same region as the primary instance, but possibly in different zones. The following figure shows the topology of read-only instances.



Read-only instances have the following features:

- Specifications of a read-only instance can be different from those of the primary instance and can be changed at any time, which facilitates elastic scaling.
- Read-only instances do not require account or database maintenance. Account and database information is synchronized with the primary instance.
- The whitelists of read-only instances can be configured independently.
- System performance monitoring is provided.
- PolarDB provides a variety of optimization recommendations, such as storage engine check, primary key check, large table check, and check for excessive indexes and missing indexes. You can optimize your databases based on the optimization recommendations and specific applications.

11.1.5.6.2. Create a read-only instance

You can create read-only instances of different specifications based on your business requirements.

Precautions

- You can create up to five read-only instances.
- Backup settings and temporary backup are not supported.
- Instance restoration is not supported.
- Data migration to read-only instances is not supported.
- Database creation and deletion are not supported.
- Account creation, deletion, authorization, and password change are not supported.
- After a read-only instance is created, you cannot restore data by directly overwriting the primary instance with a backup set.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Distributed by Instance Role** section on the right side of the **Basic Information** page, click **Create**

Read-only Instance.

5. On the **Create Read-only RDS Instance** page, configure the read-only instance parameters.

Section	Parameter	Description
Region	**Region**	The region where the PolarDB instance is located.
Specifications	**Database Type**	The database engine of the read-only instance, which is the same as that of the primary instance and cannot be changed.
	Database Version	The engine version of the read-only instance, which is the same as that of the primary instance and cannot be changed.
	Edition	Set the value to **Read-only**.
	Instance Type	The instance type of the read-only instance. The instance type of the read-only instance can be different from that of the primary instance, and can be changed at any time to facilitate flexible upgrade and downgrade. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Capacity	The storage space of the read-only instance. To ensure sufficient I/O throughput for data synchronization, we recommend that you select at least the same instance type and storage space as the primary instance for the read-only instance. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
Network Type	**Network Type**	The network type of the read-only instance, which is the same as that of the primary instance and cannot be changed.
	VPC	Select a VPC if the network type is set to VPC.
	vSwitch	Select a vSwitch if the network type is set to VPC.

6. After you configure the preceding parameters, click **Submit**.

11.1.5.6.3. View details of read-only instances

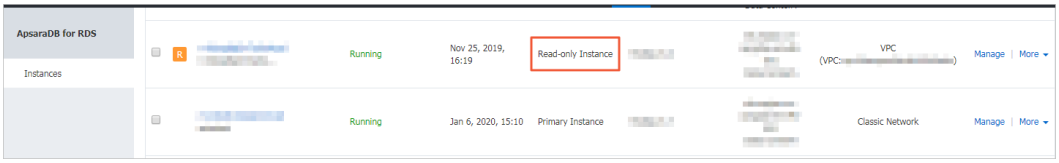
This topic describes how to view details of read-only instances. You can go to the Basic Information page of a read-only instance from the Instances page or from the read-only instance list of the primary instance. Read-only instances are managed in the same way as primary instances. The read-only instance management page shows the management operations that can be performed.

View details of a read-only instance from the Instances page

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, click the ID of a read-only instance. The **Basic Information** page appears.

   In the instance list, Instance Role of read-only instances is displayed as Read-only Instance, as shown in View a read-only instance.

   View a read-only instance



View details of a read-only instance from the Basic Information page of the primary instance

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. On the **Basic Information** page, move the pointer over the number below **Read-only Instance** in the **Distributed by Instance Role** section. The ID of the read-only instance is displayed.
5. Click the ID of the read-only instance to go to the read-only instance management page.

11.1.5.7. Change the specifications of an instance

This topic describes how to change the instance type and storage space of a PolarDB instance.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section of the **Basic Information** page, click **Change Specifications**.

   > **Notice** While you change specifications of an instance, a network interruption of about 30 seconds may occur, and most of the operations related to databases, accounts, and network operations cannot be performed. We recommend that you make appropriate arrangements for you workloads before you change specifications.

5. Change specifications based on your business requirements and click **Submit**.

11.1.5.8. Modify parameters of an instance

This topic describes how to view and modify the values of some parameters and query the parameter modification records in the console.

Modify parameters

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Parameters**.
5. You can perform the following operations:

   Export the parameter settings of the PolarDB instance to your computer.

   On the Editable Parameters tab, click **Export Parameters**. The parameter settings of the PolarDB instance are exported as a TXT file to your computer.

Modify and import the parameter settings.

i. After you modify parameters in the exported parameter file, click **Import Parameters** and copy the parameter settings to the field.

> **Note** You can select **Show Modified Parameters Only** to check the modified parameters.

ii. Click **OK**.

iii. In the upper-right corner of the page, click **Apply Changes**.

> **Note**
> - If the new parameter value takes effect only after an instance restart, the system prompts you to restart the PolarDB instance. We recommend that you restart the PolarDB instance during off-peak hours and make sure that your applications are configured with automatic reconnection policies.
> - Before the new parameter values are applied, you can click **Cancel Changes** to cancel them.

To reconfigure a single parameter, perform the following operations:

i. On the **Editable Parameters** tab, find the parameter that you want to reconfigure, and click the icon in the **Actual Value** column.

ii. Enter a new value based on the prompted value range.

iii. Click **Confirm**.

> **Note** You can select **Show Modified Parameters Only** to check the modified parameters.

iv. In the upper-right corner of the page, click **Apply Changes**.

> **Note**
> - If the new parameter value takes effect only after an instance restart, the system prompts you to restart the PolarDB instance. We recommend that you restart the PolarDB instance during off-peak hours and make sure that your applications are configured with automatic reconnection policies.
> - Before the new parameter value is applied, you can click **Cancel Changes** to cancel it.

View the parameter modification history

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Parameters**.
5. On the page that appears, click the **Edit History** tab.
6. Select a time range and then click **Search**.

11.1.6. Database connection

11.1.6.1. Connect to an instance

This topic describes how to use Data Management (DMS) or the pgAdmin 4 client to connect to an ApsaraDB RDS instance.

Context

You can log on to DMS from the ApsaraDB RDS console and then connect to an ApsaraDB RDS instance.

Data Management (DMS) is a data management service that integrates data, schema, and server management, access security, BI charts, data trends, data tracking, and performance optimization. DMS can be used to manage relational and non-relational databases, such as MySQL, SQL Server, and PostgreSQL. It can also be used to manage Linux servers.

You can also use a database client to connect to an ApsaraDB RDS instance. ApsaraDB RDS is fully compatible with PolarDB. You can connect to PolarDB instances in the similar manner as you would connect to an open source PolarDB database. In this topic, the pgAdmin 4 client is used to connect to an ApsaraDB RDS instance.

Use DMS to connect to an ApsaraDB RDS instance

For more information about how to use DMS to connect to an ApsaraDB RDS instance, see Log on to an ApsaraDB for RDS instance by using DMS.

Use the pgAdmin 4 client to connect to an ApsaraDB RDS instance

1. Add the IP address of the pgAdmin client to an IP address whitelist of the ApsaraDB RDS instance. For more information about how to configure a whitelist, see Configure an IP whitelist.
2. Start the pgAdmin 4 client.

   > **Note** For information about how to download the pgAdmin 4 client, visit pgAdmin 4 (Windows).

3. Right-click **Servers** and choose **Create > Server**, as shown in the following figure.
4. On the **General** tab of the **Create - Server** dialog box, enter the name of the server, as shown in the following figure.
5. Click the **Connection** tab and enter the information of the instance, as shown in the following figure.

Parameter	Description
Host name/address	The internal endpoint of the ApsaraDB RDS instance. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.
Port	The internal port number that is used to connect to the ApsaraDB RDS instance. For more information about how to view the internal port number, see View and modify the internal endpoint and port number.
Username	The name of the privileged account for the ApsaraDB RDS instance. For more information about how to create an privileged account, see Create a database and an account.
Password	The password of the privileged account.

6. Click **Save**.
7. If the connection information is correct, choose **Servers > Server Name > Databases > postgres**. If the following page appears, the connection is established.

> **Notice** The postgres database is the default system database of the ApsaraDB RDS instance. Do not perform operations on this database.

11.1.6.2. Configure hybrid access from both the classic network and VPCs

This topic describes how to use the hybrid access solution of ApsaraDB RDS to change the network type of an instance from classic network to Virtual Private Cloud (VPC) without network interruptions.

Prerequisites

- The network type of the ApsaraDB RDS instance is classic network.
- Available VPCs and vSwitches exist in the zone where the ApsaraDB RDS instance is deployed.

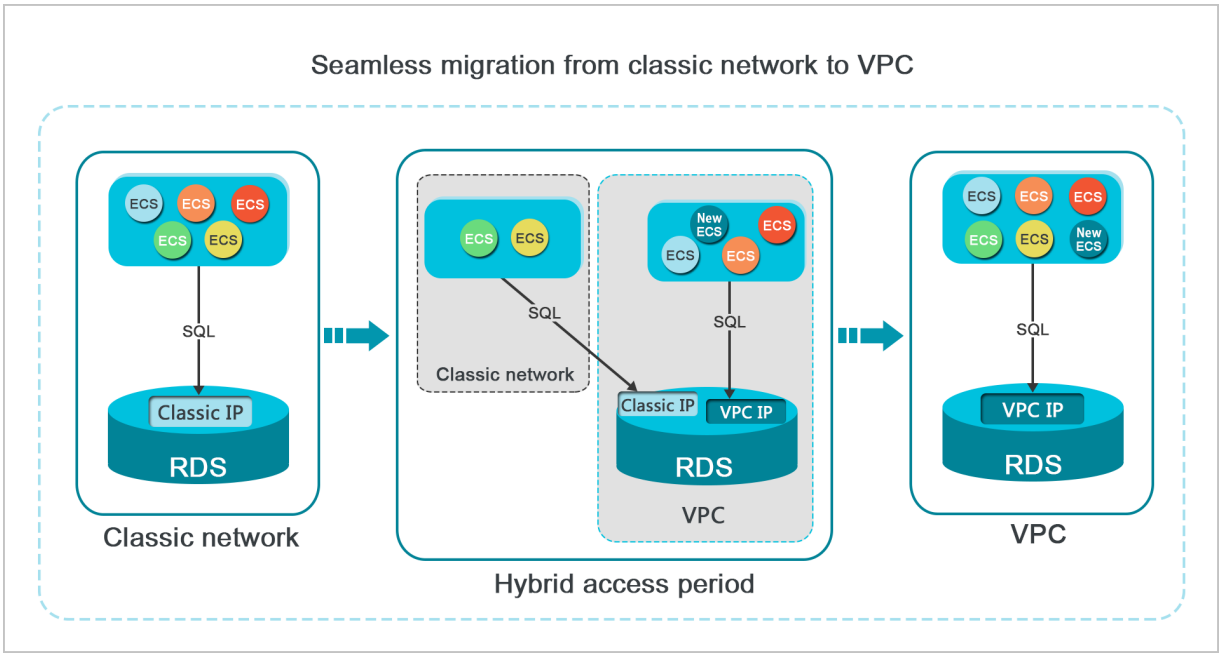
Context

In the past, when you change the network type of an ApsaraDB RDS instance from classic network to VPC, the internal endpoint of the instance would remain the same but the IP address bound to the endpoint would change to the corresponding IP address in the VPC. This change would cause a 30-second network interruption, and ECS instances within the classic network would not be able to access the ApsaraDB RDS instance by using the internal endpoint within this period. To smoothly change the network type, ApsaraDB RDS provides the hybrid access solution.

Hybrid access refers to the ability of an ApsaraDB RDS instance to be accessed by ECS instances in both the classic network and VPCs. During the hybrid access period, the ApsaraDB RDS instance reserves the original internal endpoint of the classic network and adds the internal endpoint of VPCs. This prevents network interruptions during the network type switchover.

For better security and performance, we recommend that you use the internal endpoint of VPCs. Hybrid access is available for a limited period of time. The internal endpoint of the classic network is released when the hybrid access period expires. In that case, your applications cannot access the ApsaraDB RDS database by using the internal endpoint of the classic network. You must configure the internal endpoint of VPCs in all your applications during the hybrid access period. This ensures smooth network switchover and minimizes the impact on your services.

For example, your company wants to use the hybrid access solution to change the network type from classic network to VPC. During the hybrid access period, some applications can access the database by using the internal endpoint of VPCs, and the other applications can access the database by using the original internal endpoint of the classic network. When all the applications access the database by using the internal endpoint of VPCs, the internal endpoint of the classic network can be released. The following figure illustrates the scenario.



Limits

During the hybrid access period, the instance has the following limits:

- The network type of the instance cannot be changed to classic network.
- The instance cannot be migrated to another zone.

Change the network type from classic network to VPC

For more information, see Change the network type from classic network to VPC.

Change the expiration time for the original internal endpoint of the classic network

During the period in which your instance can be accessed over the classic network or VPCs, you can specify the expiration time for the endpoint of the classic network. The setting takes effect immediately. For example, if the endpoint of the classic network expires on August 18, 2017 and you change the expiration time to 14 days later on August 15, 2017, the endpoint of the classic network is released on August 29, 2017.

To change the expiration time, perform the following operations:

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. On the **Instance Connection** tab, click **Change Expiration Time**.
6. In the **Change Expiration Time** dialog box, select an expiration time and click **OK**.

11.1.6.3. Use DMS to log on to a PolarDB instance

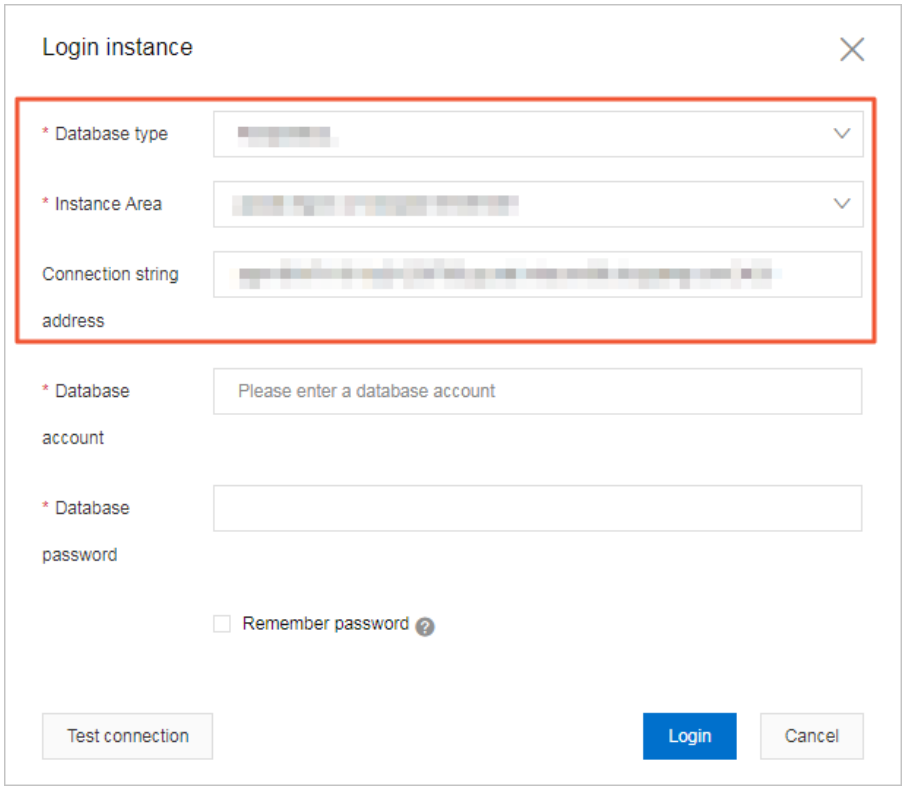
This topic describes how to use Data Management (DMS) to log on to a PolarDB instance.

Prerequisites

The IP address whitelist is configured. For more information about how to configure an IP address whitelist, see Configure an IP whitelist.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database password**, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**.

> **Note** If you want the browser to remember the password, select **Remember password** before you click **Login**.

11.1.6.4. View and modify the internal endpoint and port number

You must use the internal endpoint and port number to access a PolarDB instance. This topic describes how to view and change the internal endpoints and port numbers of a PolarDB instance in the console.

View the internal endpoint and port number

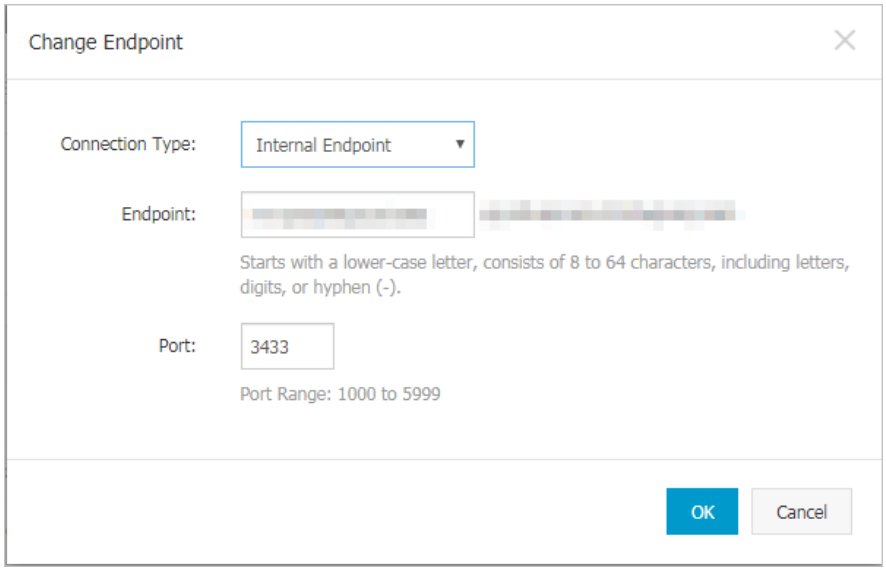
1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the **Basic Information** section, view the internal and public endpoints and port numbers.

> **Note** The endpoints and port numbers are displayed only after you configure an IP address whitelist or security group for the PolarDB instance.

Modify the internal endpoint and port number

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. On the right side of the page, click **Change Endpoint**.
6. In the dialog box that appears, set **Connection Type** to **Internal Endpoint**.
7. Modify the endpoint prefix and port number and then click **OK**.

> **Note**
>
> - The prefix of an endpoint must be 8 to 64 characters in length and can contain only letters, digits, and hyphens (-). It must start with a lowercase letter.
> - The port number must be within the range of 1000 to 5999.



FAQ

- Q: Do I need to modify the endpoint or port number in my application after I modify the endpoint or port number of an instance?

  A: Yes, you must modify the endpoint or port number in the application after you have modified them. Otherwise, the application cannot connect to databases of the instance.

- Q: Does the modification of the endpoint take effect immediately? Do I need to restart the instance?

  A: No, you do not need to restart the instance. The modification takes effect immediately.

11.1.7. Accounts

11.1.7.1. Create an account

Before you start to use ApsaraDB RDS, you must create a database for a PolarDB instance. This topic describes how to create an account for a PolarDB instance.

Precautions

- Databases within the same instance share all the resources that belong to the instance. Each PolarDB instance supports one privileged account and multiple standard accounts. You can execute SQL statements to create and manage standard accounts and databases.
- To migrate an on-premises database to a PolarDB instance, you must create a database and an account with the same names on the PolarDB instance.
- Follow the least privilege principle to create accounts and grant them appropriate read-only and read/write permissions on databases. If necessary, you can create more than one account and grant each account the permissions to access only the data within its authorized workloads. If an account does not need to write data to a database, we recommend that you grant only the read-only permissions on the database to the account.
- For security purposes, we recommend that you configure strong passwords for the accounts that are created on your PolarDB instance. In addition, we recommend that you change the passwords on a regular basis.
- After you create a privileged account for your PolarDB instance, you cannot delete the privileged account.

Create a privileged account

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.

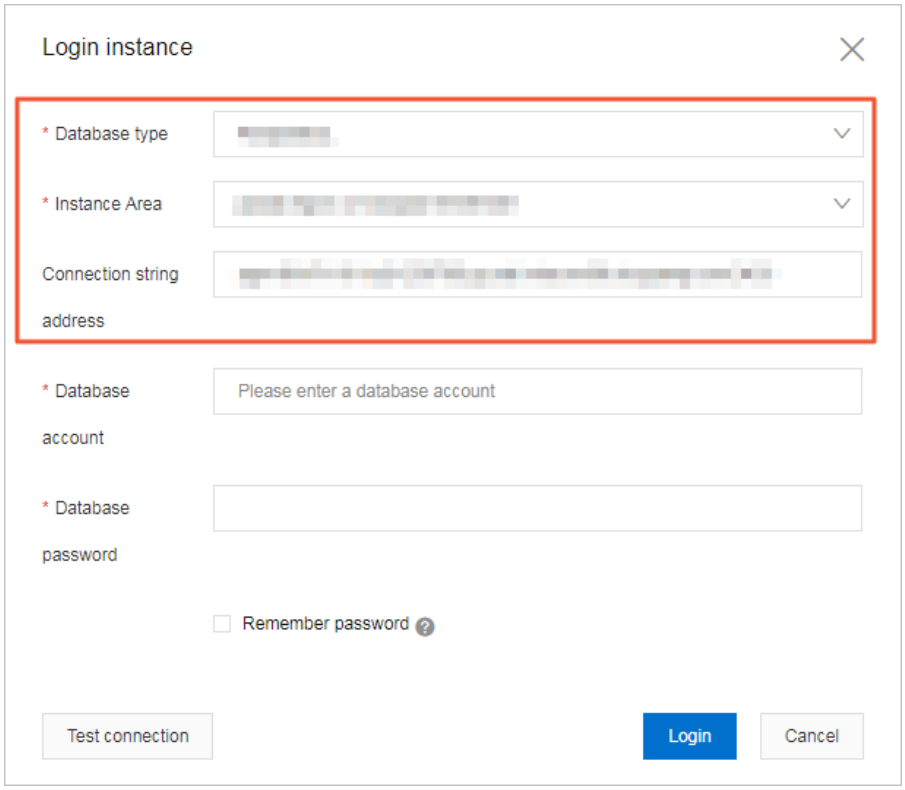
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Create Privileged Account** and configure the following parameters.

Parameter	Description
Database Account	○ The account name must be 2 to 16 characters in length. ○ The account name can contain lowercase letters, digits, and underscores (_). ○ The account name must start with a lowercase letter and end with a lowercase letter or digit.
Password	○ The password of the account must be 8 to 32 characters in length. ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ The password can contain the following characters: ! @ # $ % ^ & * ( ) _ + - =
Re-enter Password	Enter the same password again.

6. Click **Create**.

Create a standard account

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database password**, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**. If you want the browser to remember the password, select **Remember password** before you click **Login**.

> **Note** If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see Configure an IP whitelist.

7. The **SQLConsole** page appears after you log on to the instance. Execute a statement in the following format to create a standard account:

```
CREATE USER name [ [ WITH ] option [ ... ] ]
where option can be:
    SUPERUSER | NOSUPERUSER
  | CREATEDB | NOCREATEDB
  | CREATEROLE | NOCREATEROLE
  | CREATEUSER | NOCREATEUSER
  | INHERIT | NOINHERIT
  | LOGIN | NOLOGIN
  | REPLICATION | NOREPLICATION
  | CONNECTION LIMIT connlimit
  | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
  | VALID UNTIL 'timestamp'
  | IN ROLE role_name [, ...]
  | IN GROUP role_name [, ...]
  | ROLE role_name [, ...]
  | ADMIN role_name [, ...]
  | USER role_name [, ...]
  | SYSID uid
```

For example, if you want to create a user account named test2 whose password is 123456, execute the following statement:

```
create user test2 password '123456';
```

8. Click **execute**.

11.1.7.2. Reset the password

This topic describes how to use the ApsaraDB RDS console to reset the password of your database account if you forget the password.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. In the **Actions** column corresponding to the account, click **Reset Password**.
6. In the dialog box that appears, enter a new password and click **OK**.

> **Note** The password must meet the following requirements:
>
> - The password must be 8 to 32 characters in length.
> - The password must contain at least three of the following characters types: uppercase letters, lowercase letters, digits, and special characters.
> - Special characters include ! @ # $ % ^ & * ( ) _ + - =

11.1.8. Databases

11.1.8.1. Create a database

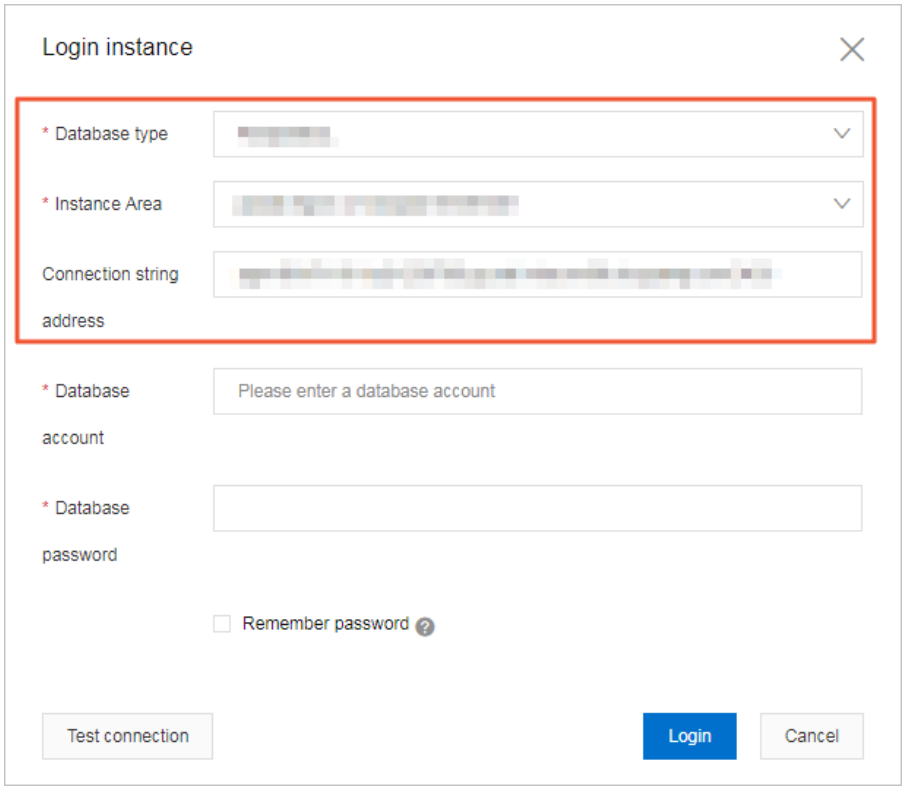
Before you start to use ApsaraDB RDS, you must create a database for an ApsaraDB RDS instance. This topic describes how to create a database on a PolarDB instance.

Prerequisites

A PolarDB instance is created. For more information, see Create an instance.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database password**, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**. If you want the browser to remember the password, select **Remember password** before you click **Login**.

> **Note** If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see Configure an IP whitelist.

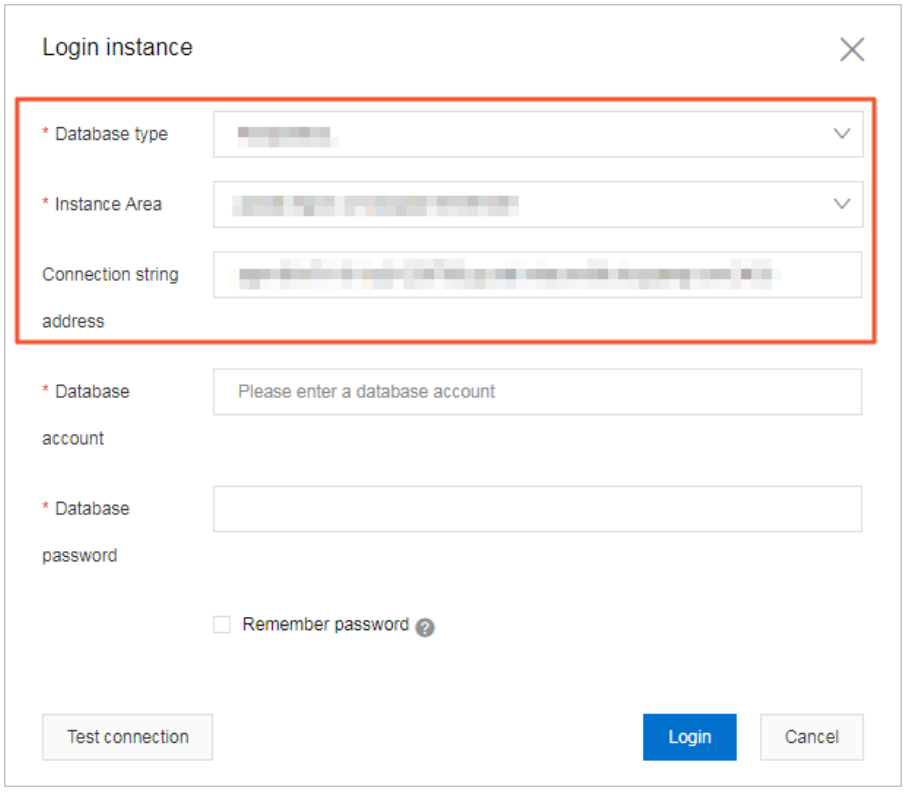
7.
8. Click **execute**.

11.1.8.2. Delete a database

This topic describes how to delete a database in the PolarDB instance.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database password**, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.

Parameter	Description
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**. If you want the browser to remember the password, select **Remember password** before you click **Login**.

   > **Note** If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see Configure an IP whitelist.

7. The **SQLConsole** page appears after you log on to the instance. Execute the following statement to delete a database:

   ```
   drop database <database name>;
   ```

8. Click **execute**.

11.1.9. Network connection

11.1.9.1. Change the network type of an instance

This topic describes how to change the network type of a PolarDB instance between classic network and VPC.

Context

- Classic network: PolarDB instances in the classic network are not isolated. You can block unauthorized access only by configuring IP address whitelists on these instances.
- VPC: Each VPC is an isolated network. A VPC provides higher security than the classic network. We recommend that you select the VPC network type.

  You can configure route tables, CIDR blocks, and gateways in a VPC. To smoothly migrate applications to the cloud, you can use leased lines or VPNs to connect your data center to a VPC to create a virtual data center.

Change the network type from VPC to classic network

Precautions

- After you change the network type from VPC to classic network, the internal endpoint of the PolarDB instance remains unchanged. However, the IP address that is associated with the internal endpoint changes.
- After you change the network type from VPC to classic network, you cannot connect Elastic Compute Service (ECS) instances deployed in VPCs to the PolarDB instance by using the internal endpoint. You must update the endpoint for the applications deployed on the ECS instances.
- When you change the network type, a 30-second network interruption may occur. To avoid business interruption, change the network type during off-peak hours or make sure that your applications are configured with automatic reconnection policies.

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.

4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the Database Connection section, click **Switch to Classic Network**.
6. In the dialog box that appears, click **OK**.

> **Note** After the network type is changed to classic network, only ECS instances within the classic network can connect to the PolarDB instance by using the internal endpoint. You must configure the internal endpoint for the ECS instances.

7. Configure a whitelist to allow ECS instances within the classic network to connect to the PolarDB instance by using the internal endpoint.

Change the network type from classic network to VPC

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the Database Connection section, click **Switch to VPC**.
6. In the Switch to VPC dialog box, select a VPC and vSwitch and specify whether to retain the endpoint used in the classic network.

> **Note**
>
> - Select a VPC. We recommend that you select the VPC where your ECS instances are deployed. Otherwise, the ECS instances cannot communicate with the PolarDB instance over the internal network.
> - Select a vSwitch. If no vSwitches are available in the selected VPC, create one in the same zone where the PolarDB instance is deployed. For more information, see Create a vSwitch in Quick Start of *VPC User Guide*.
> - Determine whether to select the **Reserve Original Classic Network Endpoint** option. The following table describes the details.
>
> - **Not selected**
>
> The classic network endpoint is not retained, and the endpoint of the instance changes to a VPC endpoint.
>
> When you change the network type, a 30-second network interruption may occur, and connections between ECS instances in the classic network and the PolarDB instance are interrupted.
>
> - **Selected**
>
> The classic network endpoint is retained, and a new VPC endpoint is generated. In such cases, the PolarDB instance runs in hybrid access mode. ECS instances in both the classic network and a VPC can connect to the PolarDB instance over an internal network.
>
> When you change the network type, no network interruptions occur. Connections between ECS instances in the classic network and the PolarDB instance remain available until the classic network endpoint expires.
>
> To migrate your business to the VPC without interruption, you must add the VPC endpoint to access the ECS instances before the classic network endpoint expires. Seven days before the classic network endpoint expires, the system sends a text message to the phone number bound to your Apsara Stack account every day.
>
> For more information, see Hybrid access from both the classic network and VPCs.

7. Add the internal IP addresses of ECS instances in the selected VPC to a VPC whitelist. This allows the ECS instances to access the PolarDB instance over the internal network. If no VPC whitelists are available, create a whitelist.

> **Note**
>
> - If you retain the classic network endpoint, add the VPC endpoint to the ECS instances before the classic network endpoint expires.
> - If you do not retain the classic network endpoint, connections between ECS instances in the classic network and the PolarDB instance over the internal network are interrupted. You must add the new endpoint to ECS instances in the VPC immediately after the network type is changed.

11.1.9.2. Configure hybrid access from both the classic network and VPCs

This topic describes how to use the hybrid access solution of ApsaraDB RDS to change the network type of an instance from classic network to Virtual Private Cloud (VPC) without network interruptions.

Prerequisites

- The network type of the ApsaraDB RDS instance is classic network.
- Available VPCs and vSwitches exist in the zone where the ApsaraDB RDS instance is deployed.

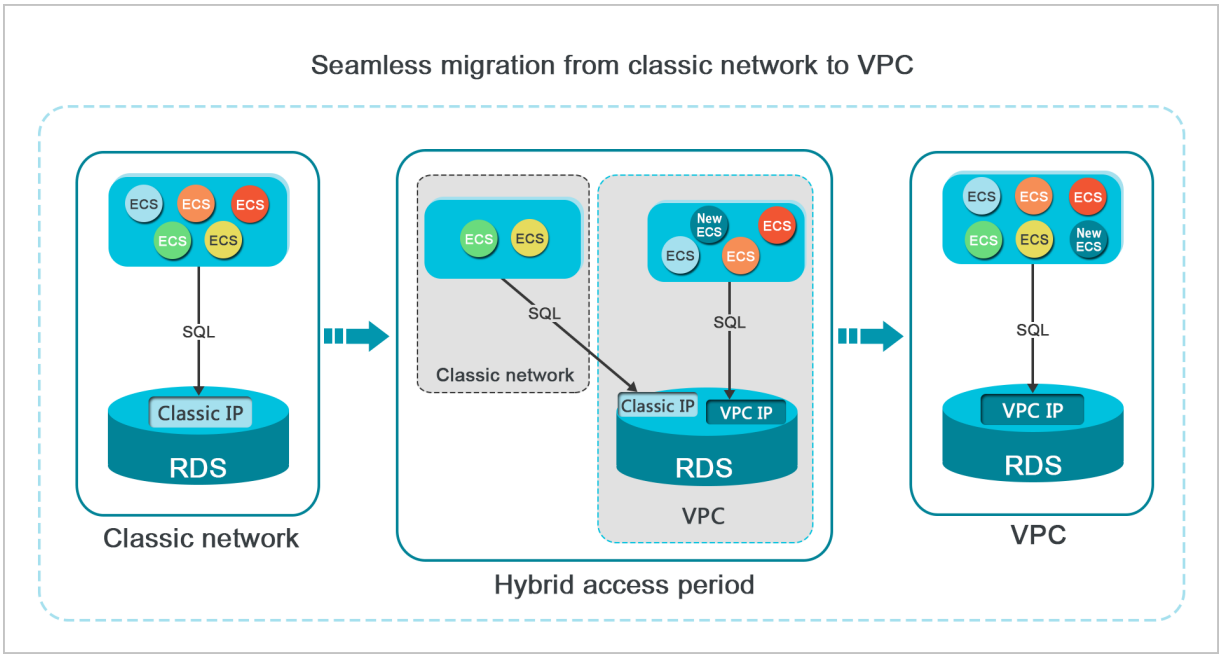
Context

In the past, when you change the network type of an ApsaraDB RDS instance from classic network to VPC, the internal endpoint of the instance would remain the same but the IP address bound to the endpoint would change to the corresponding IP address in the VPC. This change would cause a 30-second network interruption, and ECS instances within the classic network would not be able to access the ApsaraDB RDS instance by using the internal endpoint within this period. To smoothly change the network type, ApsaraDB RDS provides the hybrid access solution.

Hybrid access refers to the ability of an ApsaraDB RDS instance to be accessed by ECS instances in both the classic network and VPCs. During the hybrid access period, the ApsaraDB RDS instance reserves the original internal endpoint of the classic network and adds the internal endpoint of VPCs. This prevents network interruptions during the network type switchover.

For better security and performance, we recommend that you use the internal endpoint of VPCs. Hybrid access is available for a limited period of time. The internal endpoint of the classic network is released when the hybrid access period expires. In that case, your applications cannot access the ApsaraDB RDS database by using the internal endpoint of the classic network. You must configure the internal endpoint of VPCs in all your applications during the hybrid access period. This ensures smooth network switchover and minimizes the impact on your services.

For example, your company wants to use the hybrid access solution to change the network type from classic network to VPC. During the hybrid access period, some applications can access the database by using the internal endpoint of VPCs, and the other applications can access the database by using the original internal endpoint of the classic network. When all the applications access the database by using the internal endpoint of VPCs, the internal endpoint of the classic network can be released. The following figure illustrates the scenario.



Limits

During the hybrid access period, the instance has the following limits:

- The network type of the instance cannot be changed to classic network.
- The instance cannot be migrated to another zone.

Change the network type from classic network to VPC

For more information, see Change the network type from classic network to VPC.

Change the expiration time for the original internal endpoint of the classic network

During the period in which your instance can be accessed over the classic network or VPCs, you can specify the expiration time for the endpoint of the classic network. The setting takes effect immediately. For example, if the endpoint of the classic network expires on August 18, 2017 and you change the expiration time to 14 days later on August 15, 2017, the endpoint of the classic network is released on August 29, 2017.

To change the expiration time, perform the following operations:

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. On the **Instance Connection** tab, click **Change Expiration Time**.
6. In the **Change Expiration Time** dialog box, select an expiration time and click **OK**.

11.1.10. Monitoring

11.1.10.1. View monitoring data

PolarDB provides a wide range of performance metrics. This topic describes how to view resource monitoring data in the ApsaraDB RDS console.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. On the **Monitoring** tab, select the time range to query the corresponding monitoring data. The following table lists the specific metrics.

Metric	Description
IOPS	The number of I/O requests of the data and log disks per second.
Memory Usage	The memory usage of the instance.
CPU Utilization	The CPU utilization of the instance.

11.1.10.2. Set a monitoring frequency

This topic describes how to set the monitoring frequency of a PolarDB instance.

Context

PolarDB supports the following monitoring frequencies:

- Every 60 seconds

- Every 300 seconds

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. In the upper-right corner of the **Monitoring** tab, click **Set Monitoring Frequency**.
6. In the **Set Monitoring Frequency** dialog box, select the required monitoring frequency and click **OK**.

11.1.11. Data security

11.1.11.1. Configure an IP address whitelist

This topic describes how to configure a whitelist for a PolarDB instance. Only entities that are listed in a whitelist can access your PolarDB instance.

Context

Whitelists make your PolarDB instance more secure and do not interrupt the operations of your PolarDB instance when you configure whitelists. We recommend that you perform maintenance on your whitelists on a regular basis.

To configure a whitelist, perform the following operations:

- Configure a whitelist: Add IP addresses to allow them to connect to the PolarDB instance.

  > **Note** The default IP address whitelist contains only the IP address 127.0.0.1. This indicates that no devices are allowed to access the PolarDB instance.

- Configure an ECS security group: Add an ECS security group for the PolarDB instance to allow ECS instances in the group to connect to the PolarDB instance.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Edit** corresponding to the **default** whitelist.

   > **Note** You can also click **Create Whitelist** to create a whitelist.

6. In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks that are allowed to access the instance and click **OK**. The following section describes the rules:
   - If you enter the CIDR block 10.10.10.0/24 in the IP Addresses field, all IP addresses in the 10.10.10.X format can access your PolarDB instance.
   - If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
   - If you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all ECS instances created

within your Apsara Stack account are displayed. You can select the required IP addresses to add them to the IP address whitelist.

11.1.11.2. Configure SQL audit

This topic describes how to configure the SQL audit feature to audit SQL executions and check the details. SQL audit does not affect instance performance.

Precautions

- SQL audit does not affect instance performance.
- SQL audit logs are retained for 30 days.
- Log files exported from SQL audit are retained for two days. The system deletes files that are retained for longer than two days.
- SQL audit is disabled by default. You must manually enable it.
- You cannot view logs that are generated before SQL audit is enabled.

Enable SQL audit

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**. The **Data Security** page appears.
5. Click the **SQL Audit** tab.
6. Click **Enable SQL Audit** or **Enable now**.
7. In the message that appears, click **Confirm**.

> **Note** After SQL audit is enabled, you can query SQL information based on conditions such as the time, database, user, and keyword.

Disable SQL audit

You can disable SQL audit when it is no longer needed. To disable SQL audit, perform the following steps:

> **Notice** After SQL audit is disabled, all SQL audit logs are deleted. We recommend that you export and store audit logs to your computer before you disable SQL audit.

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**. The **Data Security** page appears.
5. Click the **SQL Audit** tab.
6. Click **Disable SQL Audit**.
7. In the message that appears, click **Confirm**.

11.1.12. Backup

11.1.12.1. Back up a PolarDB instance

This topic describes how to back up a PolarDB instance. You can configure a backup policy that is used to automatically back up your PolarDB instance. If you do not configure a backup policy, the default backup policy is used. You can also manually back up your PolarDB instance.

Precautions

- Do not perform data definition language (DDL) operations during a backup. If you do so, the backup may fail due to table locks.
- We recommend that you back up your PolarDB instance during off-peak hours.
- If your PolarDB instance has a large volume of data, a backup may require a long period of time.
- Backup files are retained for a specific retention period. We recommend that you download the required backup files to your computer before they are deleted.

Backup description

Database engine	Data backup	Log backup
PolarDB	Supports full physical backup.	Write-ahead logs (WALs) are compressed and uploaded immediately after they are generated. Each log takes up 16 MB of storage space. On-premises logs are deleted within 24 hours.

Configure a backup policy to automatically back up your PolarDB instance

ApsaraDB RDS automatically backs up your instance based on the specified backup policy.

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. On the **Backup and Restoration** page, click the **Backup Settings** tab and click **Edit**.
6. In the dialog box that appears, configure the following parameters and click **OK**. The following table lists the parameters.

Parameter	Description
Data Retention Period	The number of days for which you want to retain data backup files. Valid values: 7 to 730. Default value: 7.
Backup Cycle	The cycle to create backups. You can select one or more days of the week. **Note** To ensure data security, we recommend that you back up your PolarDB instance at least twice a week.
Backup Time	The period of time for which you want to back up data. Unit: hours.
Log backup	Specifies whether to enable the log backup feature. **Notice** If you disable this feature, all log backup files are deleted and your instance cannot be restored to previous points in time

Parameter	Description
Log Retention Period	◦ The number of days for which you want to retain log backup files. Valid values: 7 to 730. Default value: 7. ◦ The log retention period must be less than or equal to the data retention period.

Manually back up your PolarDB instance

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Back Up Instance**.
5. Select the backup mode and backup policy, and click **OK**.

   > **Note** The backup mode is **Full Backup** and the backup policy is **Instance Backup**.

6. In the upper-right corner, click the icon to view the task progress displayed in the **Task Progress** list.

   > **Note** You cannot download backup files to your computer.

FAQ

1. Q: Can I disable data backup for a PolarDB instance?

   A: No, you cannot disable the data backup feature of your PolarDB instance. However, you can reduce the backup frequency to at least twice a week. The data retention period must be within the range of 7 to 730 days.

2. Q: Can I disable log backup for a PolarDB instance?

   A: Yes, you can disable the log backup feature of your PolarDB instance. You can log on to the ApsaraDB RDS console and navigate to the Backup Settings tab to disable the log backup feature of your instance.

11.1.12.2. Download backup files

This topic describes how to download unencrypted log backup files of a PolarDB instance.

Procedure

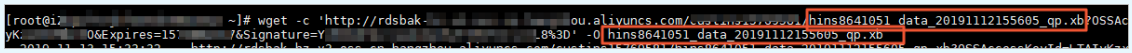
1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration** to go to the **Backup and Restoration** page.
5. Select the **Archived Logs** tab. Specify the time range to which you want to restore the instance, find the log backup file that you want to download, and then click **Download** in the **Actions** column.

> **Note** If you want to download a log backup file that is used to restore data to an on-premises database, make sure that the following requirements are met:
>
> - The instance No. of the log file must be the same as that of the data backup file.
> - The start time of the log file must be later than the data backup time and earlier than the time for restoration.

6. In the **Download Binary Log** dialog box, select a download method.

Download method	Description
Download	Download the backup file.
Copy Internal Endpoint	Copy the internal endpoint to download the file. If your ECS and PolarDB instances are deployed within the same region, you can log on to the ECS instance and use the internal endpoint to download the file. This method is fast and secure.
Copy Public Endpoint	Copy the public endpoint to download the file. If you want to use other tools to download the file, use the public endpoint.

> **Note** If you use a Linux operating system, you can run the following command to download the file:
>
> ```
> wget -c '<Public endpoint of the backup file, which is the download URL>' -O <File name>
> ```
>
> - The -c option enables resumable download.
> - The -O option saves the downloaded file by using the specified name. We recommend that you use the file name contained in the download URL.
> - If the URL contains more than one parameter, enclose the download URL in a pair of single quotation marks (').



11.1.13. Manage logs

The primary/secondary switching logs of a PolarDB instance can be used for troubleshooting. This topic describes how to manage logs of a PolarDB instance in the ApsaraDB RDS console.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Logs**.
5. On the **Logs** page, click the **Error Logs**, **Slow Query Logs**, **Slow Query Log Summary**, or **Primary/Secondary Switching Logs** tab, select a time range, and then click **Search**.

Tab	Description
Error Logs	Records database running errors that occurred within the last month.
Slow Log Details	Records SQL statements within the last month that took longer than one second to execute. Duplicated SQL statements are removed.
Slow Query Log Summary	Records and analyzes SQL statements within the last month that took longer than one second to execute. Analysis reports of slow query logs are provided.
Primary/Secondary Switching Logs	Records switchovers between the primary and secondary instances within the last month.

11.1.14. Plug-ins supported

This topic describes the plug-ins and plug-in versions supported by PolarDB.

Plug-ins and versions supported by PolarDB

Plug-in	Version
btree_gin	1.3
btree_gist	1.5
citext	1.5
cube	1.4
dict_int	1.0
earthdistance	1.1
fuzzystrmatch	1.1
hstore	1.5
intagg	1.1
intarray	1.2
isn	1.2
ltree	1.1
pg_buffercache	1.3
pg_pathman	1.5
pg_prewarm	1.2
pg_stat_statements	1.6
pg_trgm	1.4
pg_wait_sampling	1.1
pgcrypto	1.3

Plug-in	Version
pgrowlocks	1.2
pgstattuple	1.5
plpgsql	1.0
sslinfo	1.2
tablefunc	1.0
unaccent	1.1
uuid-ossp	1.1
zhparser	1.0
ganos_geometry	2.3
ganos_raster	2.3
ganos_geometry_sfcgal	2.3
ganos_geometry_topology	2.3
ganos_tiger_geocode	2.3
ganos_address_standardizer	2.3
ganos_address_standardizer_data_us	2.3
ganos_networking	2.3
ganos_pointcloud	2.3
ganos_trajectory	2.3
plperl	1.0
pltcl	1.0

Note PolarDB updates its engine version to support new plug-ins or plug-in versions. To view the supported plug-ins, execute the following statement:

```
show polar_supported_extensions;
```

11.1.15. PolarDB driver

11.1.15.1. Download the driver

This topic describes how to download the drivers that are used to connect to PolarDB instances.

You can download and install a driver in this topic. The driver allows your application to access a PolarDB instance.

Drivers

- JDBC:

  polardb_oracle_jdbc.zip

  For more information about how to use the PolarDB Java Database Connectivity (JDBC) driver to connect to a PolarDB instance, see RDS PolarDB JDBC.

- .NET:

  polardb_oracle_.net.zip

  For more information about how to use the PolarDB .NET driver to connect to a PolarDB instance, see RDS PolarDB .NET.

- OCI:

  polardb_oracle_oci.tar

  For more information about how to use the Oracle Call Interface (OCI) driver to connect to a PolarDB instance, see RDS PolarDB OCI.

- ODBC:

  polardb_oracle_odbc.tar.gz

  For more information about how to use the Open Database Connectivity (ODBC) driver to connect to a PolarDB instance, see RDS PolarDB ODBC.

11.1.15.2. RDS PolarDB JDBC

This topic describes how to use the PolarDB Java Database Connectivity (JDBC) driver to connect Java applications to PolarDB instances.

Prerequisites

An account is created for a PolarDB instance. For more information, see Create an account.

Context

JDBC is a Java API that is used to connect Java applications to databases. PolarDB JDBC is developed based on open source PostgreSQL JDBC. PolarDB JDBC uses PostgreSQL protocols for LAN communications. PolarDB JDBC allows Java applications to connect to databases by using standard and database-independent Java code.

The PolarDB JDBC driver uses version 3.0 of the PostgreSQL protocol, and is compatible with Java 6 (JDBC 4.0), Java 7 (JDBC 4.1), and Java 8 (JDBC 4.2).

Download the package of the PolarDB JDBC driver

Download the package of the PolarDB JDBC driver. Apsara Stack provides three Java versions of the PolarDB JDBC driver. If you select the version that is compatible with Java 6, use the `polardb-jdbc16.jar` JAR file. If you select the version that is compatible with Java 7, use the `polardb-jdbc17.jar` JAR file. If you select the version that is compatible with Java 8, use the `polardb-jdbc18.jar` JAR file. You can select a version of the PolarDB JDBC driver based on the Java Development Kit (JDK) version that is used by your application.

Configure the PolarDB JDBC driver

Before you use the PolarDB JDBC driver in your Java application, add the path where the JDBC driver package is stored to `CLASSPATH`. For example, if your JDBC driver is stored in */usr/local/polardb/share/java/*, run the following command to add the JDBC driver path to `CLASSPATH`:

```
export CLASSPATH=$CLASSPATH:/usr/local/polardb/share/java/<Name of the JAR file.jar>
```

Example:

```
export CLASSPATH=$CLASSPATH:/usr/local/polardb/share/java/polardb-jdbc18.jar
```

Run the following command to view the current JDBC version:

```
#java -jar <Name of the JAR file.jar>
```

Example:

```
#java -jar polardb-jdbc18.jar
POLARDB JDBC Driver 42.2.5.2.0
```

Set up a Java project by using Maven

If your Java project is set up by using Maven, run the following command to install the PolarDB JDBC driver package in your repository:

```
 mvn install:install-file -DgroupId=com.aliyun -DartifactId=<Name of the JAR file> -Dversion=1.1.2 -D
packaging=jar -Dfile=/usr/local/polardb/share/java/<Name of the JAR file.jar>
```

Example:

```
 mvn install:install-file -DgroupId=com.aliyun -DartifactId=polardb-jdbc18 -Dversion=1.1.2 -Dpackagin
g=jar -Dfile=/usr/local/polardb/share/java/polardb-jdbc18.jar
```

Add the following dependency to the `pom.xml` file of the Maven project:

```
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId><Name of the JAR file></artifactId>
    <version>1.1.2</version>
</dependency>
```

Example:

```
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>polardb-jdbc18</artifactId>
    <version>1.1.2</version>
</dependency>
```

Hibernate

In the `hibernate.cfg.xml` Hibernate configuration file, configure the driver class and dialect of the PolarDB database if your project uses Hibernate to connect to the database.

> **Note** Only Hibernate 3.6 and later support PostgresPlusDialect.

```
<property name="connection.driver_class">com.aliyun.polardb.Driver</property>
<property name="connection.url">jdbc:polardb://pc-***.o.polardb.rds.aliyuncs.com:1521/polardb_test</p
roperty>
<property name="dialect">org.hibernate.dialect.PostgresPlusDialect</property>
```

Druid connection pool

When you use the Druid connection pool, specify the `driver name` and `dbtype` parameters in an explicit manner, as shown in the following example:

```
dataSource.setDriverClassName("com.aliyun.polardb.Driver");
dataSource.setDbType("polardb");
```

> **Note** For versions earlier than Druid 1.1.22, set the `dbType` parameter to `postgresql`.

Modify configurations to adapt to Activiti

If your application uses the Activiti framework for business process management (BPM), the following error message may appear when you initialize PolarDB data sources:

```
couldn't deduct database type from database product name 'POLARDB Database Compatible with Oracle'
```

The reason is that Activiti provides built-in mappings between database versions and database types. However, in the built-in mappings, the database version is not mapped to PolarDB. To resolve this issue, you can specify the subclass of `SpringProcessEngineConfiguration` and reload `buildProcessEngine` to the subclass. You must specify the database types in an explicit manner. The following example shows how to specify the database types:

```
package com.aliyun.polardb;
import org.activiti.engine.ProcessEngine;
import org.activiti.spring.SpringProcessEngineConfiguration;
public class PolarDBSpringProcessEngineConfiguration extends SpringProcessEngineConfiguration {
    public PolarDBSpringProcessEngineConfiguration() {
        super();
    }
    @Override
    public ProcessEngine buildProcessEngine() {
        setDatabaseType(DATABASE_TYPE_POSTGRES);
        return super.buildProcessEngine();
    }
}
```

Save the subclass of `SpringProcessEngineConfiguration` in your project and use the subclass in the configuration file to load configurations. Then, initialize the engine. The following example provides details:

```
<bean id="processEngineConfiguration" class="com.aliyun.polardb.PolarDBSpringProcessEngineConfiguration">
      <property name="dataSource" ref="dataSource"/>
      <property name="transactionManager" ref="transactionManager"/>
      <property name="databaseSchemaUpdate" value="true"/>
      <!-- Other configurations are omitted. -->
</bean>
```

Load the PolarDB JDBC driver

```
Class.forName("com.aliyun.polardb.Driver");
```

Example

```
package com.aliyun.polardb;
import java.sql.Connection;
import java.sql.Driver;
```

```
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Properties;
/**
 * POLARDB JDBC DEMO
 * <p>
 * Please make sure the host ip running this demo is in you cluster's white list.
 */
public class PolarDBJdbcDemo {
  /**
   * Replace the following information.
   */
  private final String host = "***.o.polardb.rds.aliyuncs.com";
  private final String user = "***";
  private final String password = "***";
  private final String port = "1921";
  private final String database = "db_name";
  public void run() throws Exception {
    Connection connect = null;
    Statement statement = null;
    ResultSet resultSet = null;
    try {
      Class.forName("com.aliyun.polardb.Driver");
      Properties props = new Properties();
      props.put("user", user);
      props.put("password", password);
      String url = "jdbc:polardb://" + host + ":" + port + "/" + database;
      connect = DriverManager.getConnection(url, props);
      /**
       * create table foo(id int, name varchar(20));
       */
      String sql = "select id, name from foo";
      statement = connect.createStatement();
      resultSet = statement.executeQuery(sql);
      while (resultSet.next()) {
        System.out.println("id:" + resultSet.getInt(1));
        System.out.println("name:" + resultSet.getString(2));
      }
    } catch (Exception e) {
      e.printStackTrace();
      throw e;
    } finally {
      try {
        if (resultSet != null)
          resultSet.close();
        if (statement != null)
          statement.close();
        if (connect != null)
          connect.close();
      } catch (SQLException e) {
        e.printStackTrace();
        throw e;
      }
    }
  }
  public static void main(String[] args) throws Exception {
    PolarDBJdbcDemo demo = new PolarDBJdbcDemo();
```

```
    demo.run();
  }
}
```

In JDBC, a database is typically represented by a URL, as shown in the following example:

```
jdbc:polardb://pc-***.o.polardb.rds.aliyuncs.com:1521/polardb_test?user=test&password=Pw123456
```

Parameter	Example	Description
URL prefix	`jdbc:polardb://`	The prefix of the URL. All the prefixes of the PolarDB instance URLs are `jdbc:polardb://` .
Endpoint	`pc-***.o.polardb.rds.aliyuncs.com`	The endpoint of the PolarDB instance. For more information about how to view the endpoint, see View or apply for an endpoint in *PolarDB User Guide*.
Port number	`1521`	The port of the PolarDB instance. Default value: 1521.
Database	`polardb_test`	The name of the database to be connected to.
Account	`test`	The username that is used to connect to the PolarDB instance.
Password	`Pw123456`	The password for the username of the PolarDB instance.

When you perform a query on a database, you must create a `Statement` , `PreparedStatment` , or `CallableStatement` object.

In the preceding example, a `Statement` object is created. A `PreparedStatment` object is created in the following example:

```
PreparedStatement st = conn.prepareStatement("select id, name from foo where id > ?");
st.setInt(1, 10);
resultSet = st.executeQuery();
while (resultSet.next()) {
    System.out.println("id:" + resultSet.getInt(1));
    System.out.println("name:" + resultSet.getString(2));
}
```

The following example shows how to use `CallableStatement` to process a stored procedure:

```
String sql = "{?=call getName (?, ?, ?)}";
CallableStatement stmt = conn.prepareCall(sql);
stmt.registerOutParameter(1, java.sql.Types.INTEGER);
//Bind IN parameter first, then bind OUT parameter
int id = 100;
stmt.setInt(2, id); // This would set ID as 102
stmt.registerOutParameter(3, java.sql.Types.VARCHAR);
stmt.registerOutParameter(4, java.sql.Types.INTEGER);
//Use execute method to run stored procedure.
stmt.execute();
//Retrieve name with getXXX method
String name = stmt.getString(3);
Integer msgId = stmt.getInt(4);
Integer result = stmt.getInt(1);
System.out.println("Name with ID:" + id + " is " + name + ", and messegeID is " + msgId + ", and retu
rn is " + result);
```

The following code is used to create the stored procedure `getName` that is used in the preceding example:

```
CREATE OR REPLACE FUNCTION getName(
    id        In      Integer,
    name      Out     Varchar2,
    result    Out     Integer
  ) Return Integer
Is
  ret     Int;
Begin
  ret := 0;
  name := 'Test';
  result := 1;
  Return(ret);
End;
```

11.1.15.3. RDS PolarDB .NET

This topic describes how to use the ADO.NET Data Provider for PolarDB (PolarDB .NET) driver to connect C# applications to a PolarDB instance.

Prerequisites

An account is created for a PolarDB instance. For more information, see Create an account.

Context

PolarDB .NET is a driver that allows you to use programming languages such as C#, Visual Basic, and F# to connect to PolarDB databases. The PolarDB .NET driver is compatible with Entity Framework Core and Entity Framework 6.x. You can use PolarDB .NET in conjunction with Entity Framework to develop applications in a quick way.

The current driver uses the PostgreSQL 3.0 protocol and is compatible with .NET Framework 4.x and .NET Core 2.x.

Entity Framework overview

Entity Framework is an object-relational mapping (ORM) framework that is widely adopted in the .NET platform. Entity Framework and Language-Integrated Query (LINQ) technologies provide a quick method for you to develop backend C# applications.

The PolarDB .NET driver provides the .dll files of Entity Framework 5 (EF5) and Entity Framework 6 (EF6), and the .dll files are applicable to PolarDB. This helps you use Entity Framework.

For more information about Entity Framework, visit the official website of Entity Framework.

Install the PolarDB .NET driver

1. Download the package of the POLARDB .NET driver.
2. Decompress the package of the PolarDB .NET driver.

```
unzip polardb_oracle_.net.zip
```

3. Import the driver to the Visual Studio project.

   Add the following content to the <Project> node of sample. csproj or the GUI of Visual Studio:

```
<Project>
  ...
  <ItemGroup>
    <Reference Include="POLARDB.POLARDBClient, Version=4.0.4.1, Culture=neutral, PublicKeyToken=5
d8b90d52f46fda7">
      <HintPath>${your path}\POLARDB.POLARDBClient.dll</HintPath>
    </Reference>
  </ItemGroup>
  ...
</Project>
```

Example

In the Samples folder, you can see the polardb-sample.sql file and multiple sample project files. The following procedure shows how to run these sample projects.

1. Connect to a database. For more information, see Connect to an instance.
2. Run the following command to create a project named `sampledb`:

```
CREATE DATABASE sampledb;
```

3. Import the databases, tables, data, and functions that are required for testing to database `sampledb`.

```
\i ${your path}/polardb-sample.sql
```

4. After the data is imported, write C # code.

   The following sample code shows how to query, update, and call stored procedures:

```
using System;
using System.Data;
using POLARDB.POLARDBClient;
/*
 * This class provides a simple way to perform DML operation in POLARDB
 *
 * @revision 1.0
 */
namespace POLARDBClientTest
{
    class SAMPLE_TEST
    {
        static void Main(string[] args)
        {
            POLARDBConnection conn = new POLARDBConnection("Server=localhost;Port=5432;User Id=po
laruser;Password=password;Database=sampledb");
            try
            {
                conn.Open();
                //Simple select statement using POLARDBCommand object
```

```
                POLARDBCommand POLARDBSeletCommand = new POLARDBCommand("SELECT EMPNO,ENAME,JOB,M
GR,HIREDATE FROM EMP",conn);
                POLARDBDataReader SelectResult =  POLARDBSeletCommand.ExecuteReader();
                while (SelectResult.Read())
                {
                    Console.WriteLine("Emp No" + " " + SelectResult.GetInt32(0));
                    Console.WriteLine("Emp Name" + " " + SelectResult.GetString(1));
                    if (SelectResult.IsDBNull(2) == false)
                        Console.WriteLine("Job" + " " + SelectResult.GetString(2));
                    else
                        Console.WriteLine("Job" + " null ");
                    if (SelectResult.IsDBNull(3) == false)
                        Console.WriteLine("Mgr" + " " + SelectResult.GetInt32(3));
                    else
                        Console.WriteLine("Mgr" + "null");
                    if (SelectResult.IsDBNull(4) == false)
                        Console.WriteLine("Hire Date" + " " + SelectResult.GetDateTime(4));
                    else
                        Console.WriteLine("Hire Date" + " null");
                    Console.WriteLine("--------------------------------");
                }
                //Insert statement using POLARDBCommand Object
                SelectResult.Close();
                POLARDBCommand POLARDBInsertCommand = new POLARDBCommand("INSERT INTO EMP(EMPNO,E
NAME) VALUES((SELECT COUNT(EMPNO) FROM EMP),'JACKSON')",conn);
                POLARDBInsertCommand.ExecuteScalar();
                Console.WriteLine("Record inserted");
                //Update  using POLARDBCommand Object
                POLARDBCommand  POLARDBUpdateCommand = new POLARDBCommand("UPDATE EMP SET ENAME =
'DOTNET' WHERE EMPNO < 100",conn);
                POLARDBUpdateCommand.ExecuteNonQuery();
                Console.WriteLine("Record has been updated");
                POLARDBCommand POLARDBDeletCommand = new POLARDBCommand("DELETE FROM EMP WHERE EM
PNO < 100",conn);
                POLARDBDeletCommand.CommandType= CommandType.Text;
                POLARDBDeletCommand.ExecuteScalar();
                Console.WriteLine("Record deleted");
                //procedure call example
                try
                {
                    POLARDBCommand callable_command = new POLARDBCommand("emp_query(:p_deptno,:p_
empno,:p_ename,:p_job,:p_hiredate,:p_sal)", conn);
                    callable_command.CommandType = CommandType.StoredProcedure;
                    callable_command.Parameters.Add(new POLARDBParameter("p_deptno",POLARDBTypes.
POLARDBDbType.Numeric,10,"p_deptno",ParameterDirection.Input,false ,2,2,System.Data.DataRowVersio
n.Current,20));
                    callable_command.Parameters.Add(new POLARDBParameter("p_empno", POLARDBTypes.
POLARDBDbType.Numeric,10,"p_empno",ParameterDirection.InputOutput,false ,2,2,System.Data.DataRowV
ersion.Current,7369));
                    callable_command.Parameters.Add(new POLARDBParameter("p_ename", POLARDBTypes.
POLARDBDbType.Varchar,10,"p_ename",ParameterDirection.InputOutput,false ,2,2,System.Data.DataRowV
ersion.Current,"SMITH"));
                    callable_command.Parameters.Add(new POLARDBParameter("p_job", POLARDBTypes.PO
LARDBDbType.Varchar,10,"p_job",ParameterDirection.Output,false ,2,2,System.Data.DataRowVersion.Cu
rrent,null));
                    callable_command.Parameters.Add(new POLARDBParameter("p_hiredate", POLARDBTyp
es.POLARDBDbType.Date,200,"p_hiredate",ParameterDirection.Output,false ,2,2,System.Data.DataRowVe
rsion.Current,null));
                    callable_command.Parameters.Add(new POLARDBParameter("p_sal", POLARDBTypes.PO
```

```
LARDBDbType.Numeric,200,"p_sal",ParameterDirection.Output,false ,2,2,System.Data.DataRowVersion.C
urrent,null));
                    callable_command.Prepare();
                    callable_command.Parameters[0].Value = 20;
                    callable_command.Parameters[1].Value = 7369;
                    POLARDBDataReader result = callable_command.ExecuteReader();
                    int fc = result.FieldCount;
                    for(int i=0;i<fc;i++)
                        Console.WriteLine("RESULT["+i+"]="+ Convert.ToString(callable_command.Par
ameters[i].Value));
                    result.Close();
                }
                catch(POLARDBException exp)
                {
                    if(exp.ErrorCode.Equals("01403"))
                        Console.WriteLine("No data found");
                    else if(exp.ErrorCode.Equals("01422"))
                        Console.WriteLine("More than one rows were returned by the query");
                    else
                        Console.WriteLine("There was an error Calling the procedure. \nRoot Cause
:\n");
                    Console.WriteLine(exp.Message.ToString());
                }
                //Prepared statement
                string updateQuery  = "update emp set ename = :Name where empno = :ID";
                POLARDBCommand Prepared_command = new POLARDBCommand(updateQuery, conn);
                Prepared_command.CommandType = CommandType.Text;
                Prepared_command.Parameters.Add(new POLARDBParameter("ID", POLARDBTypes.POLARDBDb
Type.Integer));
                Prepared_command.Parameters.Add(new POLARDBParameter("Name", POLARDBTypes.POLARDB
DbType.Text));
                Prepared_command.Prepare();
                Prepared_command.Parameters[0].Value = 7369;
                Prepared_command.Parameters[1].Value = "Mark";
                Prepared_command.ExecuteNonQuery();
                Console.WriteLine("Record Updated...");
            }
            catch(POLARDBException exp)
            {
                Console.WriteLine(exp.ToString() );
            }
            finally
            {
                conn.Close();
            }
        }
    }
}
```

Connection string parameters

Applications must provide connection strings to connect to databases. The connection strings include the host, username, and password parameters.

Connection strings are in the `keyword1=value; keyword2=value;` format, and are case-insensitive. You can use double quotation marks (") to enclose the values that contain special characters, such as semicolons (;).

The current driver supports the following connection string parameters.

Parameter	Example	Description
Host	`localhost`	The endpoint of the PolarDB instance. For more information about how to view the endpoint, see View or apply for an endpoint in *PolarDB User Guide*.
Port	`1521`	The port of the PolarDB instance. Default value: 1521.
Database	`sampledb`	The name of the database to be connected.
Username	`polaruser`	The username that is used to connect to the PolarDB instance.
Password	`password`	The password for the username of the PolarDB instance.

Other parameters.

Parameter	Description
application_name	The name of the application.
search_path	The search path of the schema.
client_encoding	The client encoding.
timezone	The time zone of the session.

11.1.15.4. RDS PolarDB ODBC

This topic describes how to use the PolarDB Open Database Connectivity (ODBC) driver to connect a Unix or Linux application to a PolarDB instance.

Prerequisites

An account is created for a PolarDB instance. For more information, see Create an account.

Download and install the ODBC driver

1. Download the driver.
2. Install the driver.

   Note PolarDB provides an ODBC driver package. You can decompress the package and use the ODBC driver without the need to install it. Run the following command to decompress the package:

   ```
   tar -zxvf polardb_oracle_odbc.tar.gz
   ```

Connect to PolarDB

1. Install Libtool on the Linux server. Libtool must be version 1.5.1 or later.

   ```
   yum install -y libtool
   ```

2. Install unixODBC-devel on the Linux server.

   ```
   yum install -y unixODBC-devel
   ```

3. Edit the `odbcinst.ini` file in the */etc* directory.

```
vim /etc/odbcinst.ini
```

4. Add the following content to the `odbcinst.ini` file:

```
[POLARDB]
Description = ODBC for POLARDB
Driver      = /root/target/lib/unix/polar-odbc.so
Setup       = /root/target/lib/unix/libodbcpolarS.so
Driver64    = /root/target/lib/unix/polar-odbc.so
Setup64     = /root/target/lib/unix/libodbcpolarS.so
Database    = <Name of the database>
Servername  = <Endpoint of the PolarDB instance>
Password    = <Password>
Port        = <Port>
Username    = <Username>
Trace       = yes
TraceFile   = /tmp/odbc.log
FileUsage   = 1
```

> **Note**
> - For more information about how to view the endpoint of a PolarDB instance, see View and modify the internal endpoint and port number.
> - In the preceding sample code, replace */root* with the actual path of the `target` folder.

5. Connect to PolarDB.

```
$isql -v POLARDB
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
SQL>
```

Examples

The following examples show how to run the `Test1` and `Test2` files.

1. Open the `samples` folder in the ODBC driver folder.

```
cd samples
```

2. Compile the sample test. The test files named `Test1` and `Test2` are generated.

```
make
```

3. Run `Test1` and `Test2`.

```
./Test1
## Run Test1
./Test2
## Run Test2
```

> **Note**
>
> - `Test1` contains the example to perform add, delete, update, and query operations. `Test2` contains the example of cursors and output parameters.
> - The following sample code is only a snippet of the source code. To view the complete source code, you can check the `Test1` and `Test2` files in the `samples` folder of the ODBC driver package.

Sample code of Test1:

```
...
int main(int argc, char* argv[])
{
    /*Initialization*/
    RETCODE rCode;
    HENV *hEnv = (HENV*)malloc(sizeof(HENV));
    HDBC *hDBC = (HDBC*)malloc(sizeof(HDBC));
    HSTMT *hStmt = (HSTMT*)malloc(sizeof(HSTMT));
    Connect("POLARDB","user","",&hEnv,&hDBC);
    rCode = SQLAllocStmt(*hDBC,hStmt);
    rCode = SQLAllocHandle(SQL_HANDLE_STMT,*hDBC,hStmt);
    /*Add, delete, update, and query operations*/
    ExecuteInsertStatement(&hStmt,(UCHAR*) "INSERT INTO EMP(EMPNO,ENAME) VALUES((SELECT COUNT(EMPNO)
FROM EMP),'JACKSON')");
    ExecuteUpdate(&hStmt,(UCHAR*) "UPDATE EMP SET ENAME='ODBC Test' WHERE EMPNO < 100");
    ExecuteDeletStatement(&hStmt,(UCHAR*) "DELETE FROM EMP WHERE EMPNO<100");
    ExecuteSimple_Select(&hStmt,(UCHAR*) "SELECT EMPNO,ENAME,JOB,MGR,HIREDATE FROM EMP where empno =
7369");
    /*Disconnection*/
    Disconnect(&hEnv,&hDBC,&hStmt);
    /*clean up*/
    free(hEnv);
    free(hDBC);
    free(hStmt);
    return 0;
}
```

Sample code for Test2:

```
int main(int argc, char* argv[])
{
    /*Definition*/
    RETCODE rCode;
    SQLUSMALLINT a;
    SQLINTEGER Num1IndOrLen;
    SQLSMALLINT iTotCols = 0;
    int j;
    SDWORD cbData;
    /*Initialization*/
    HENV *hEnv = (HENV*)malloc(sizeof(HENV));
    HDBC *hDBC = (HDBC*)malloc(sizeof(HDBC));
    HSTMT *hStmt = (HSTMT*)malloc(sizeof(HSTMT));
    HSTMT *hStmt1 = (HSTMT*)malloc(sizeof(HSTMT));
    /**Establish a connection**/
    Connect("POLARDB","user","***",&hEnv,&hDBC);
    rCode = SQLAllocStmt(*hDBC,hStmt);
    rCode = SQLAllocStmt(*hDBC,hStmt1);
    rCode = SQLAllocHandle(SQL_HANDLE_STMT,*hDBC,hStmt);
    rCode = SQLAllocHandle(SQL_HANDLE_STMT,*hDBC,hStmt1);
    /*begin*/
    ExecuteSimple_Select(&hStmt1,(UCHAR*) "BEGIN;");
    /*prepare*/
    RETCODE rc = SQLPrepare((*hStmt),(SQLCHAR*)"{ call refcur_inout_callee2(?,?)}",SQL_NTS);
    rc = SQLBindParameter((*hStmt),1, SQL_PARAM_INPUT_OUTPUT, SQL_C_CHAR,SQL_REFCURSOR,0, 31,
                strName, 31, &Num1IndOrLen);
    rc = SQLBindParameter((*hStmt),2, SQL_PARAM_INPUT_OUTPUT, SQL_C_CHAR,SQL_REFCURSOR,0, 31,
                    &strName1, 31, &Num1IndOrLen);
    Num1IndOrLen=0;
    /*execute*/
    rc = SQLExecute((*hStmt));
    if(rc == SQL_SUCCESS || rc == SQL_SUCCESS_WITH_INFO)
    {
        printf("\nstrName _________ = %s\n",strName);
        printf("\nstrName 1_________ = %s\n",strName1);
    }
    printf("\n First Cursor as OUT Parameter \n")   ;
```

11.1.15.5. RDS PolarDB OCI

This topic describes how to use the PolarDB Oracle Call Interface (OCI) driver to connect to PolarDB.

Prerequisites

- An account is created on a PolarDB instance. For more information, see Create an account.
- The server operating system is 64-bit Linux or Windows.
- The development kit of the Oracle OCI driver is installed.

Context

PolarDB OCI is the native C language interface to databases of PolarDB. You can use PolarDB OCI to build other language-specific interfaces, including PolarDB Java Database Connectivity (JDBC), PolarDB .Net, and PolarDB Open Database Connectivity (ODBC). It allows you to execute query statements and make SQL function calls for PolarDB databases.

The current driver version is PostgreSQL 3.0.

Download the PolarDB OCI driver

polardb-oci.tar.gz

Install the PolarDB OCI driver

Decompress the driver package and import the driver files to environment variables. This allows you to find the correct location of the driver when you compile a demo.

You can use the following methods to manually import the driver files to the environment variables in Linux and Windows:

- Linux
  i. Copy the libpolaroci.so.10.2, libiconv.so.2, and libpq.so.5.11 files to the */usr/lib* directory.
  ii. Create a symbolic link.

```
ln -s /usr/lib/libpolaroci.so.10.2 /usr/lib/libpolaroci.so
ln -s /usr/lib/libiconv.so.2 /usr/lib/libiconv.so
ln -s /usr/lib/libpq.so.5.11 /usr/lib/libpq.so
ln -s /usr/lib/libpq.so.5.11 /usr/lib/libpq.so.5
```

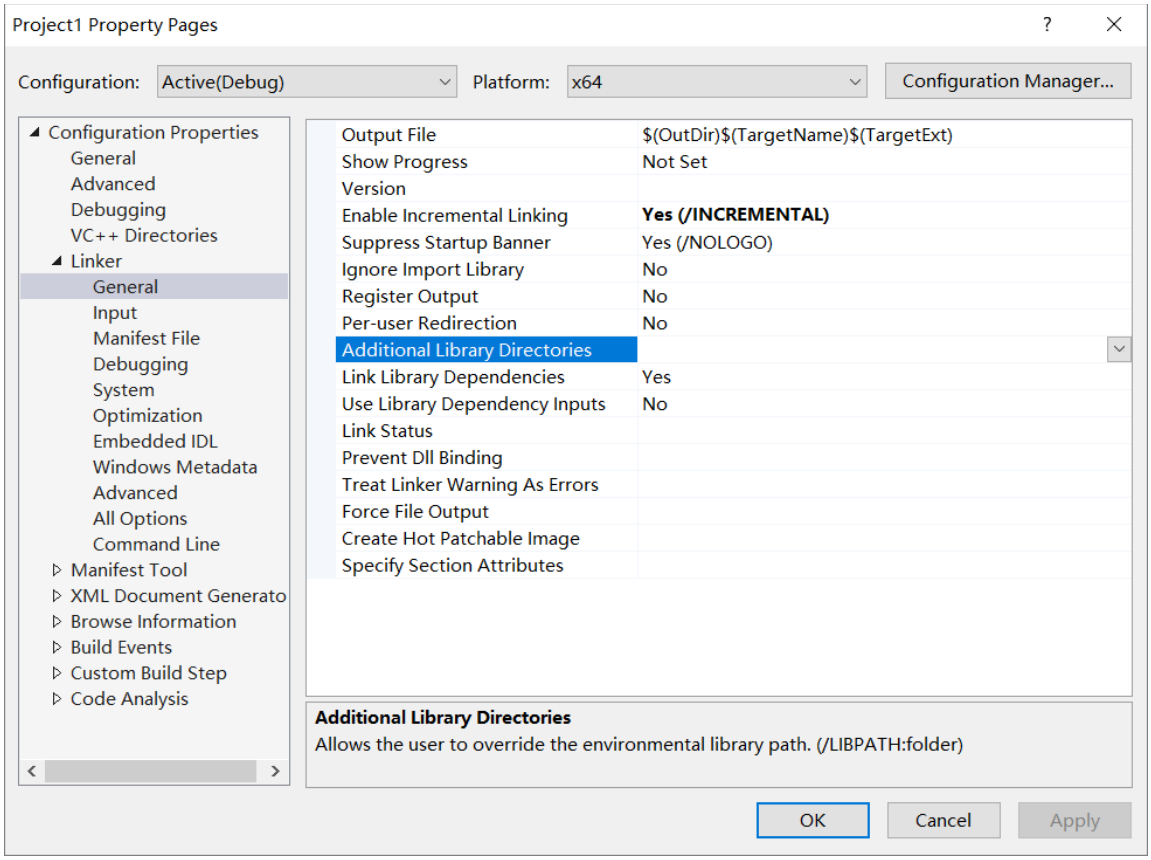
  iii. Set environment variables in Linux.

```
export LD_LIBRARY_PATH= /usr/lib
```

  > **Note**
  > - If the libiconv.so files already exist in your system, you can directly use these files. You can also follow the instructions in the libiconv documentation to download, compile, and install libiconv, and then use the compiled .so files.
  > - In Linux, the libiconv.so files provided by the PolarDB OCI driver are for reference only.

- Windows
  i. Set environment variables.

  The IDE editor in Windows is generally capable of importing the paths of linked files. In this topic, Visual Studio is used to illustrate how to import the paths of linked files, as shown in the following figure.



ii. On the properties tab of the project, add additional library directories. Then, add the .dll files in the driver directory to **Additional Library Directories**.

Write code

The following code example shows how to write code. The demo `polardb_demo` in the *sample* directory is used in the example.

```
/* ================================================================================
 * Copyright (c) 2004-2019 POLARDB Corporation. All Rights Reserved.
 * ================================================================================
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <oci.h>
#ifdef WIN32
#include <time.h>
#else
#include <sys/time.h>
#endif
/* Define a macro to handle errors */
#define HANDLE_ERROR(x,y) check_oci_error(x,y)
#define DATE_FMT "DAY, MONTH DD, YYYY"
#define DATE_LANG "American"
sword ConvertStringToDATE( char *datep, char *formatp, dvoid *datepp );
/* A Custom Routine to handle errors,     */
/* this demonstrates the Error/ Exception Handling in OCI */
void check_oci_error (dvoid * errhp, sword status);
/*
 * <<<<<<<<<<<<<<<<<<< FUNCTION PROTOTYPES<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
 */
```

```
 */
/* Initialize    & Allocate all handles */
void
initHandles (OCISvcCtx **, OCIServer **, OCISession **, OCIError **,
         OCIEnv **);
/* logon to the database and begin user-session */
void
logon (OCISvcCtx **, OCIServer **, OCISession **, OCIError **,
       OCIEnv **, text *, text *, text *);
/* Create required table(s)  */
void create_table (OCISvcCtx *, OCIError *, OCIEnv *);
/* prepare data for our examples */
void prepare_data (OCISvcCtx *, OCIError *, OCIEnv *);
/* create procedures/functions to demonstrate in the example */
void create_stored_procs (OCISvcCtx *, OCIError *, OCIEnv *);
/* select and print data by iterating through resultSet */
void select_print_data (OCISvcCtx *, OCIError *, OCIEnv *);
/* demonstrate calling stored procedures and retrieving values */
/* proc1 demonstrates IN OUT */
void call_stored_proc1 (OCISvcCtx *, OCIError *, OCIEnv *);
/* proc2 demonstrates OUT */
void call_stored_proc2 (OCISvcCtx *, OCIError *, OCIEnv *);
/* drop required table(s) */
void drop_table (OCISvcCtx *, OCIError *, OCIEnv *);
/* drop stored procedures and functions */
void drop_stored_procs (OCISvcCtx *, OCIError *, OCIEnv *);
/* clean-up main handles before exit */
void
cleanup (OCISvcCtx **, OCIServer **, OCISession **, OCIError **, OCIEnv **);
/*
 * <<<<<<<<<<<<<<<<<<<<<<<< END OF FUNCTION PROTOYPES<<<<<<<<<<<<<<<<<<<<<<<<
 */
/* <<<<<<<<<< Global Variables */
ub4 init_mode = OCI_DEFAULT;
ub4 auth_mode = OCI_CRED_RDBMS;
/* <<<<<<<<<< End Global Variables */
int
main (void)
{
  /*
   * Declare Handles, a typical OCI program would need atleast
   * following handles Enviroment Handle Error Handle Service Context
   * Handle Server Handle User Session (Authentication Handle)
   */
  /* Enviroment */
  OCIEnv *envhp;
  /* Error */
  OCIError *errhp;
  /* Service Context */
  OCISvcCtx *svchp;
  /* Server */
  OCIServer *srvhp;
  /* Session(authentication) */
  OCISession *authp;
  /*
   * End of Handle declaration
   */
  /*
   * Declare local variables,
   */
```

```
  text *username = (text *) "parallels";
  text *passwd = (text *) "";
  /*
   * Oracle Instant Client Connection String
   */
  text *server = (text *) "//localhost:5432/postgres";
  /*
   * Initialize and Allocate handles
   */
  initHandles (&svchp, &srvhp, &authp, &errhp, &envhp);
  /*
   * logon to the database
   */
  logon (&svchp, &srvhp, &authp, &errhp, &envhp, username, passwd, server);
  /*
   * Create table(s) required for this example
   */
  create_table (svchp, errhp, envhp);
  /*
   * insert data into table
   */
  prepare_data (svchp, errhp, envhp);
  /*
   * create stored procedures & functions
   */
  create_stored_procs (svchp, errhp, envhp);
  /*
   * select and print data by iterating through simple resultSet
   */
  select_print_data (svchp, errhp, envhp);
  /*
   * demonstrate calling stored procedures and retrieving values
   */
  call_stored_proc1 (svchp, errhp, envhp);
  /*
   * demonstrate OUT parameters
   */
  call_stored_proc2 (svchp, errhp, envhp);
  /*
   * Drop table(s) used in this example
   */
  drop_table (svchp, errhp, envhp);
  /*
   * Drop stroed procedures & functions used in this example
   */
  drop_stored_procs (svchp, errhp, envhp);
  /*
   * clean up resources
   */
  cleanup (&svchp, &srvhp, &authp, &errhp, &envhp);
  return 0;
}
/* A Custom Routine to handle errors,     */
/* this demonstrates the Error/ Exception Handling in OCI */
void
check_oci_error (dvoid * errhp, sword status)
{
  text errbuf[512];
  sb4 errcode;
```

```
  if (status == OCI_SUCCESS)
    {
      return;
    }
  switch (status)
    {
    case OCI_SUCCESS_WITH_INFO:
      printf ("OCI_SUCCESS_WITH_INFO:\n");
      OCIErrorGet (errhp, (ub4) 1, (text *) 0, &errcode,
           errbuf, (ub4) sizeof (errbuf), OCI_HTYPE_ERROR);
      printf ("%s", errbuf);
      break;
    case OCI_NEED_DATA:
      printf ("Error - OCI_NEED_DATA\n");
      break;
    case OCI_NO_DATA:
      printf ("Error - OCI_NO_DATA\n");
      break;
    case OCI_ERROR:
      printf ("Error - OCI_ERROR:\n");
      OCIErrorGet (errhp, (ub4) 1, (text *) 0, &errcode,
           errbuf, (ub4) sizeof (errbuf), OCI_HTYPE_ERROR);
      printf ("%s", errbuf);
      break;
    case OCI_INVALID_HANDLE:
      printf ("Error - OCI_INVALID_HANDLE\n");
      break;
    case OCI_STILL_EXECUTING:
      printf ("Error - OCI_STILL_EXECUTING\n");
      break;
    case OCI_CONTINUE:
      printf ("Error - OCI_CONTINUE\n");
      break;
    default:
      break;
    }
  /*
   * exit app
   */
  exit((int)status);
}
/* Initialize & Allocate required handles */
void
initHandles (OCISvcCtx ** svchp, OCIServer ** srvhp, OCISession ** authp,
         OCIError ** errhp, OCIEnv ** envhp)
{
  /*
   * Now Starts the Section where we have to initialize & Allocate
   * basic handles. This is a compulsory setup or initilization which
   * is required before we can proceed to logon and work with the
   * database. This initialization and prepration will include the
   * following steps
   *
   * 1. Initialize the OCI (OCIInitialize()) 2. Initialize the
   * Environment (OCIEnvInit()) 3. Initialize & Allocate Error Handle
   * 4. Initialize & Allocate Service Context Handle 5. Initialize &
   * Allocate Session Handle 6. Initialize & Allocate Server Handle
   *
   * As per the new versions of OCI , instead of using OCIInitialize()
   * and OCIEnvInit(), we can do this with one API Call called
```

```
   * OCIEnvCreate().
   */
  /*
   * Initialize OCI
   */
  if (OCIInitialize (init_mode, (dvoid *) 0,
              (dvoid * (*)(dvoid *, size_t)) 0,
              (dvoid * (*)(dvoid *, dvoid *, size_t)) 0,
              (void (*)(dvoid *, dvoid *)) 0) != OCI_SUCCESS)
    {
      printf ("ERROR: failed to initialize OCI\n");
      exit (1);
    }
  /*
   * Initialize Enviroment.
   */
  HANDLE_ERROR (*envhp,
        OCIEnvInit (&(*envhp), OCI_DEFAULT, (size_t) 0,
                (dvoid **) 0));
  /*
   * Initialize & Allocate Error Handle
   */
  HANDLE_ERROR (*envhp,
        OCIHandleAlloc (*envhp, (dvoid **) & (*errhp),
                OCI_HTYPE_ERROR, (size_t) 0, (dvoid **) 0));
  /*
   * Initialize & Allocate Service Context Handle
   */
  HANDLE_ERROR (*errhp,
        OCIHandleAlloc (*envhp, (dvoid **) & (*svchp),
                OCI_HTYPE_SVCCTX, (size_t) 0, (dvoid **) 0));
  /*
   * Initialize & Allocate Session Handle
   */
  HANDLE_ERROR (*errhp,
        OCIHandleAlloc (*envhp, (dvoid **) & (*authp),
                OCI_HTYPE_SESSION, (size_t) 0, (dvoid **) 0));
  /*
   * Initialize & Allocate Server Handle
   */
  HANDLE_ERROR (*errhp,
        OCIHandleAlloc (*envhp, (dvoid **) & (*srvhp),
                OCI_HTYPE_SERVER, (size_t) 0, (dvoid **) 0));
}
void
logon (OCISvcCtx ** svchp, OCIServer ** srvhp, OCISession ** authp,
       OCIError ** errhp, OCIEnv ** envhp, text * username, text * passwd,
       text * server)
{
  /*
   * Now Starts our Logon to the Database Server which includes two
   * steps
   *
   * 1. Attaching to the Server 2. Starting or Begining of the Session
   *
   * This is the complex logon. The easy ways to logon is to avoid
   * server attach and session begin and simply use OCILogon() or
   * OCILogon2() and then logoff using OCILogoff()
   */
```

```
  /*
   * Attach to the server
   */
  HANDLE_ERROR (*errhp,
        OCIServerAttach (*srvhp, *errhp, server,
                 (ub4) strlen ((char *) server),
                 OCI_DEFAULT));
  /*
   * The following code will start a session but before we start a
   * session we have to 1. Set the Server Handle which is now attached
   * into Service Context Handle 2. Set the Username and password into
   * Session Handle
   */
  /*
   * Set the Server Handle into Service Context Handle
   */
  HANDLE_ERROR (*errhp,
        OCIAttrSet (*svchp, OCI_HTYPE_SVCCTX,
                (dvoid *) (*srvhp), (ub4) 0, OCI_ATTR_SERVER,
                *errhp));
  /*
   * Set the username and password into session handle
   */
  HANDLE_ERROR (*errhp,
        OCIAttrSet (*authp, OCI_HTYPE_SESSION,
                (dvoid *) username,
                (ub4) strlen ((char *) username),
                OCI_ATTR_USERNAME, *errhp));
  HANDLE_ERROR (*errhp,
        OCIAttrSet (*authp, OCI_HTYPE_SESSION, (dvoid *) passwd,
                (ub4) strlen ((char *) passwd), OCI_ATTR_PASSWORD,
                *errhp));
  /*
   * Now FINALLY Begin our session
   */
  HANDLE_ERROR ((*errhp),
        OCISessionBegin (*svchp, *errhp,
                 *authp, auth_mode, OCI_DEFAULT));
  printf ("*******************************************\n");
  printf ("Milestone  : Logged on as --> '%s'\n", username);
  printf ("*******************************************\n");
  /*
   * After we Begin our session we will have to set the Session
   */
  /*
   * (authentication) handle into Service Context Handle
   */
  HANDLE_ERROR (*errhp,
        OCIAttrSet (*svchp, OCI_HTYPE_SVCCTX,
                (dvoid *) (*authp), (ub4) 0,
                OCI_ATTR_SESSION, *errhp));
}
/* Create table(s) required for this example */
void
create_table (OCISvcCtx * svchp, OCIError * errhp, OCIEnv * envhp)
{
  OCIStmt *stmhp;
  text *create_statement =
    (text *)"CREATE TABLE OCISPEC \n (ENAME VARCHAR2(20)\n, MGR NUMBER\n, HIREDATE DATE)";
  ub4 status = OCI_SUCCESS;
```

```
  /*
   * Initialize & Allocate Statement Handle
   */
  HANDLE_ERROR (errhp,
        OCIHandleAlloc (envhp, (dvoid **) & stmhp,
                OCI_HTYPE_STMT, (size_t) 0, (dvoid **) 0));
  /*
   * Prepare the Create statement
   */
  HANDLE_ERROR (errhp,
        OCIStmtPrepare (stmhp, errhp,
                create_statement,
                strlen ((const char *) create_statement),
                OCI_NTV_SYNTAX, OCI_DEFAULT));
  /*
   * Execute the Create Statement
   */
  if ((status = OCIStmtExecute (svchp, stmhp, errhp,
                  (ub4) 1, (ub4) 0, NULL, NULL, OCI_DEFAULT)) < OCI_SUCCESS)
    {
      printf ("FAILURE IN CREATING TABLE(S)\n");
      HANDLE_ERROR (errhp, status);
      return;
    }
  else
    {
      printf ("*******************************************\n");
      printf ("MileStone : Table(s) Successfully created\n");
      printf ("*******************************************\n");
    }
  HANDLE_ERROR (errhp, OCIHandleFree (stmhp, OCI_HTYPE_STMT));
}
/* prepare data for our examples */
void
prepare_data (OCISvcCtx * svchp, OCIError * errhp, OCIEnv * envhp)
{
  OCIStmt *stmhp;
  text *insstmt =
    (text *)
    "INSERT INTO OCISPEC (ename,mgr, hiredate) VALUES (:ENAME,:MGR, CAST(:HIREDATE AS timestamp))";
  OCIBind *bnd1p = (OCIBind *) 0;    /* the first bind handle   */
  OCIBind *bnd2p = (OCIBind *) 0;    /* the second bind handle */
  OCIBind *bnd3p = (OCIBind *) 0;    /* the third bind handle   */
  ub4 status = OCI_SUCCESS;
  int i = 0;
  char *ename[3] = { "SMITH", "ALLEN", "KING" };
  sword mgr[] = { 7886, 7110, 7221 };
  char *date_buffer[3] = { "02-AUG-07", "02-APR-07", "02-MAR-07" };
  /*
   * Initialize & Allocate Statement Handle
   */
  HANDLE_ERROR (errhp,
        OCIHandleAlloc (envhp, (dvoid **) & stmhp,
                OCI_HTYPE_STMT, (size_t) 0, (dvoid **) 0));
  /*
   * Prepare the insert statement
   */
  HANDLE_ERROR (errhp,
        OCIStmtPrepare (stmhp, errhp, insstmt,
```

```
                (ub4) strlen ((char *) insstmt),
                (ub4) OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
  /*
   * In this loop we will bind data from the arrays to insert multi
   * rows in the database a more elegant and better way to do this is
   * to use Array Binding (Batch Inserts). POLARDB OCI Replacement
   * Library WILL support Array Bindings even if it is not used here
   * right now
   */
  for (i = 0; i < 3; i++)
    {
      /*
       * Bind Variable for ENAME
       */
      HANDLE_ERROR (errhp,
            OCIBindByName (stmhp, &bnd1p, errhp, (text *) ":ENAME",
                  -1, (dvoid *) ename[i],
                  (sb4) strlen (ename[i]) + 1, SQLT_STR,
                  (dvoid *) 0, 0, (ub2 *) 0, (ub4) 0,
                  (ub4 *) 0, OCI_DEFAULT));
      /*
       * Bind Variable for MGR
       */
      HANDLE_ERROR (errhp,
            OCIBindByName (stmhp, &bnd2p, errhp, (text *) ":MGR",
                  -1, (dvoid *) & mgr[i], sizeof (mgr[i]),
                  SQLT_INT, (dvoid *) 0, 0, (ub2 *) 0,
                  (ub4) 0, (ub4 *) 0, OCI_DEFAULT));
      /*
       * Bind Variable for HIREDATE
       */
      HANDLE_ERROR (errhp,
            OCIBindByName (stmhp, &bnd3p, errhp, (text *) ":HIREDATE",
                  -1, (dvoid *)  date_buffer[i],
                  strlen(date_buffer[i])+1, SQLT_STR, (dvoid *) 0, 0,
                  (ub2 *) 0, (ub4) 0, (ub4 *) 0,
                  OCI_DEFAULT));
      /*
       * Execute the statement and insert data
       */
      if ((status = OCIStmtExecute (svchp, stmhp, errhp,
                  (ub4) 1, (ub4) 0, NULL, NULL, OCI_DEFAULT)) < OCI_SUCCESS)
    {
      printf ("FAILURE IN INSERTING DATA\n");
      HANDLE_ERROR (errhp, status);
      return;
    }
    }
  OCITransCommit (svchp, errhp, (ub4) 0);
  printf ("*******************************************\n");
  printf
    ("MileStone : Data Sucessfully inserted \n & Committed via Transaction\n");
  printf ("*******************************************\n");
  HANDLE_ERROR (errhp, OCIHandleFree (stmhp, OCI_HTYPE_STMT));
}
/* Create Stored procedures and functions to be used in this example */
void
create_stored_procs (OCISvcCtx * svchp, OCIError * errhp, OCIEnv * envhp)
{
  /*
```

```
   * This function created 2 stored procedures and one stored function
   * 1. StoredProcedureSample1 - is to exhibit exeucting procedure and
   * recieving values from an IN OUT parameter 2.
   * StoredProcedureSample2 - is to exhibit executing procedure and
   * recieving values from an OUT parameter 3. StoredProcedureSample3 -
   * is to exhibit executing a function and recieving the value
   * returned by the function in a Callable Statement way
   */
  OCIStmt *stmhp;
  OCIStmt *stmhp2;
  OCIStmt *stmhp3;
  text *create_statement =
    (text *)"CREATE OR REPLACE PROCEDURE StoredProcedureSample1\n (mgr1 int, ename1 IN OUT varchar2)\
n   is\nbegin\ninsert into ocispec (mgr, ename) values (7990,'STOR1');\nename1 := 'Successful';\n end
;\n";
  text *create_statement2 =
    (text *)"CREATE OR REPLACE PROCEDURE StoredProcedureSample2\n(mgr1 int, ename1 varchar2,eout1 OUT
varchar2)\nis\nbegin\ninsert into ocispec(mgr,ename) values (7991, 'STOR2');\neout1 := 'Successful';\
n    end;";
  text *create_statement3 =
    (text *)"CREATE OR REPLACE FUNCTION f1\nRETURN VARCHAR2\nis\nv_Sysdate DATE;\nv_charSysdate VARCH
AR2(20);\nbegin\nSELECT TO_CHAR(SYSDATE, 'dd-mon-yyyy') into v_charSysdate FROM DUAL;\n    return(v_c
harSysdate);\nend;";
  ub4 status = OCI_SUCCESS;
  /*
   * Initialize & Allocate Statement Handles
   */
  HANDLE_ERROR (errhp,
        OCIHandleAlloc (envhp, (dvoid **) & stmhp,
                OCI_HTYPE_STMT, (size_t) 0, (dvoid **) 0));
  HANDLE_ERROR (errhp,
        OCIHandleAlloc (envhp, (dvoid **) & stmhp2,
                OCI_HTYPE_STMT, (size_t) 0, (dvoid **) 0));
  HANDLE_ERROR (errhp,
        OCIHandleAlloc (envhp, (dvoid **) & stmhp3,
                OCI_HTYPE_STMT, (size_t) 0, (dvoid **) 0));
  /*
   * Prepare the Create statements
   */
  HANDLE_ERROR (errhp,
        OCIStmtPrepare (stmhp, errhp,
                create_statement,
                strlen ((const char *) create_statement),
                OCI_NTV_SYNTAX, OCI_DEFAULT));
  HANDLE_ERROR (errhp,
        OCIStmtPrepare (stmhp2, errhp, create_statement2,
                strlen ((const char *) create_statement2),
                OCI_NTV_SYNTAX, OCI_DEFAULT));
  HANDLE_ERROR (errhp,
        OCIStmtPrepare (stmhp3, errhp, create_statement3,
                strlen ((const char *) create_statement3),
                OCI_NTV_SYNTAX, OCI_DEFAULT));
  /*
   * Execute the Create Statement SampleProcedure1
   */
  if ((status = OCIStmtExecute (svchp, stmhp, errhp,
                  (ub4) 1, (ub4) 0, NULL, NULL, OCI_DEFAULT)) < OCI_SUCCESS)
    {
      printf ("FAILURE IN CREATING PROCEDURE 1\n");
```

```
      HANDLE_ERROR (errhp, status);
      return;
    }
  else
    {
      printf ("*********************************************\n");
      printf ("MileStone : Sample Procedure 1 Successfully created\n");
      printf ("*********************************************\n");
    }
  /*
   * Execute the Create Statement Sample Procedure2
   */
  if ((status = OCIStmtExecute (svchp, stmhp2, errhp,
                  (ub4) 1, (ub4) 0, NULL, NULL, OCI_DEFAULT)) < OCI_SUCCESS)
    {
      printf ("FAILURE IN CREATING PROCEDURE 2\n");
      HANDLE_ERROR (errhp, status);
      return;
    }
  else
    {
      printf ("*********************************************\n");
      printf ("MileStone : Sample Procedure 2 Successfully created\n");
      printf ("*********************************************\n");
    }
  /*
   * Execute the Create Statement Sample Procedure3
   */
  if ((status = OCIStmtExecute (svchp, stmhp3, errhp,
                  (ub4) 1, (ub4) 0, NULL, NULL, OCI_DEFAULT)) < OCI_SUCCESS)
    {
      printf ("FAILURE IN CREATING PROCEDURE 3\n");
      HANDLE_ERROR (errhp, status);
      return;
    }
  else
    {
      printf ("*********************************************\n");
      printf ("MileStone : Sample Procedure 3 Successfully created\n");
      printf ("*********************************************\n");
    }
  HANDLE_ERROR (errhp, OCIHandleFree (stmhp, OCI_HTYPE_STMT));
  HANDLE_ERROR (errhp, OCIHandleFree (stmhp2, OCI_HTYPE_STMT));
  HANDLE_ERROR (errhp, OCIHandleFree (stmhp3, OCI_HTYPE_STMT));
}
/* select and print data by iterating through resultSet */
void
select_print_data (OCISvcCtx * svchp, OCIError * errhp, OCIEnv * envhp)
{
  /* Statement */
  OCIStmt *stmhp;
  /* Define */
  OCIDefine *define;
  /* Buffer for employee Name */
  char ename_buffer[10] ;
  /* Buffer for mgr */
  sword mgr_buffer;
  /*Buffer for hiredate */
  char hire_date[20];
```

```
  /*
   * a simple select statement
   */
  text * sql_statement =
      (text *) "select ename,mgr,hiredate from ocispec";
  /*
   * additional local variables
   */
  ub4 rows = 1;
  ub4 fetched = 1;
  ub4 status = OCI_SUCCESS;
  sb2 null_ind_ename = 0;
  /* null indicator for ename */
  sb2 null_ind_mgr = 0;
  /* null indicator for mgr */
  sb2 null_ind_hiredate = 0;
  /* null indicator for hiredate */
  /*
   * Now we are going to start the Milestone of a Simple Query of the
   * database and loop through the resultSet This would include
   * following steps
   *
   * 1. Initialize and Allocate the Statement Handle 2. Prepare the
   * Statement 3. Define Output variables to recieve the output of the
   * select statement 4. Execute the statement 5. Fetch the resultset
   * and Print values
   *
   */
memset( ename_buffer, 0, sizeof(ename_buffer) );
memset( hire_date, 0, sizeof(hire_date) );
  /*
   * Initialize & Allocate Statement Handle
   */
  HANDLE_ERROR (errhp,
        OCIHandleAlloc (envhp, (dvoid **) & stmhp,
                OCI_HTYPE_STMT, (size_t) 0, (dvoid **) 0));
  /*
   * Prepare the statement
   */
  HANDLE_ERROR (errhp,
        OCIStmtPrepare (stmhp, errhp,
                sql_statement,
                strlen ((const char *) sql_statement),
                OCI_NTV_SYNTAX, OCI_DEFAULT));
  /*
   * Bind a String (OCIString) variable on position 1. Datatype used
   * SQLT_VST
   */
  HANDLE_ERROR (errhp,
        OCIDefineByPos (stmhp, &define, errhp,
                (ub4) 1, ename_buffer, 10,
                (ub2) SQLT_STR, &null_ind_ename, 0, 0,
                OCI_DEFAULT));
  /*
   * Bind a Number (OCINumber) variable on position 2. Datatype used
   * SQLT_VNU
   */
  HANDLE_ERROR (errhp,
        OCIDefineByPos (stmhp, &define, errhp,
                (ub4) 2, &mgr buffer, sizeof (sword),
```

```
                (ub2) SQLT_INT, &null_ind_mgr, 0, 0,
                OCI_DEFAULT));
/*
 * Bind a Date (OCIDate) variable on position 3. Datatype used
 * SQLT_ODT
 */
HANDLE_ERROR (errhp,
      OCIDefineByPos (stmhp, &define, errhp,
                (ub4) 3, hire_date, 20,
                (ub2) SQLT_STR, &null_ind_hiredate, 0, 0,
                OCI_DEFAULT));
/*
 * Execute the simple SQL Statement
 */
status = OCIStmtExecute (svchp, stmhp, errhp,
             rows, (ub4) 0, NULL, NULL, OCI_DEFAULT);
/*
 * Print the Resultset
 */
if (status == OCI_NO_DATA)
  {
    /*
     * indicates didn't fetch anything (as we're not array
     * fetching)
     */
    fetched = 0;
  }
else
  {
    HANDLE_ERROR (errhp, status);
  }
if (fetched)
  {
    /*
     * print string
     */
    if (null_ind_ename == -1)
  printf ("name -> [NULL]\t");
    else
  printf ("name -> [%s]\t",  ename_buffer);
    /*
     * print number by converting it into int
     */
    if (null_ind_mgr == -1)
  printf ("mgr -> [NULL]\n");
    else
  {
    printf ("mgr -> [%d]\n", mgr_buffer);
  }
    if (null_ind_hiredate == -1)
  printf ("hiredate -> [NULL]\n");
    else
  {
    printf ("hiredate -> [%s]\n",hire_date );
  }
    /*
     * loop through the resultset one by one through
     * OCIStmtFetch()
     */
```

```
      /*
       * untill we find nothing
       */
      while (1)
    {
      status = OCIStmtFetch (stmhp, errhp,
                rows, OCI_FETCH_NEXT, OCI_DEFAULT);
      if (status == OCI_NO_DATA)
        {
          /*
           * indicates couldn't fetch anything
           */
          break;
        }
      else
        {
          HANDLE_ERROR (errhp, status);
        }
      /*
       * print string
       */
      if (null_ind_ename == -1)
        printf ("name -> [NULL]\t");
      else
        printf ("name -> [%s]\t", ename_buffer);
      /*
       * print number by converting it into int
       */
      if (null_ind_mgr == -1)
        printf ("mgr -> [NULL]\n");
      else
        {
        printf ("mgr -> [%d]\n", mgr_buffer);
        }
      /*
       * print date after converting to text
       */
      if (null_ind_hiredate == -1)
        printf ("hiredate -> [NULL]\n");
      else
        {
          printf ("hiredate -> [%s]\n", hire_date);
        }
    }
    }
  HANDLE_ERROR (errhp, OCIHandleFree (stmhp, OCI_HTYPE_STMT));
}
void
call_stored_proc1 (OCISvcCtx * svchp, OCIError * errhp, OCIEnv * envhp)
{
  OCIStmt *p_sql;
  OCIBind *p_Bind1 = (OCIBind *) 0;
  OCIBind *p_Bind2 = (OCIBind *) 0;
  char field2[20];
  /*
   * char field3[20];
   */
  sword field1 = 3;
  text *mySql = (text *) "Begin StoredProcedureSample1(:MGR, :ENAME); END";
  memset( field2, 0, sizeof(field2) );
```

```
  strcpy( field2, "Entry 3" );
  printf ("*************************************************\n");
  printf ("Example 1 - Using an IN OUT Parameter\n");
  printf ("*************************************************\n");
  /*
   * Initialize & Allocate Statement Handle
   */
  HANDLE_ERROR (errhp,
        OCIHandleAlloc (envhp, (dvoid **) & p_sql,
                OCI_HTYPE_STMT, (size_t) 0, (dvoid **) 0));
  HANDLE_ERROR (errhp,
        OCIStmtPrepare (p_sql, errhp, mySql,
                (ub4) strlen ((char *)mySql), OCI_NTV_SYNTAX,
                OCI_DEFAULT));
  HANDLE_ERROR (errhp,
        OCIBindByPos (p_sql, &p_Bind1, errhp, 1,
                  (dvoid *) & field1, sizeof (sword),
                  SQLT_INT, 0, 0, 0, 0, 0, OCI_DEFAULT));
  HANDLE_ERROR (errhp,
        OCIBindByPos (p_sql, &p_Bind2, errhp, 2,
                  field2, (sizeof (field2)),
                  SQLT_STR, 0, 0, 0, 0, 0, OCI_DEFAULT));
  printf (" Field2 Before:\n");
  printf (" size ---> %d\n", sizeof (field2));
  printf (" length ---> %d\n", strlen (field2));
  printf (" value ---> %s\n", field2);
  HANDLE_ERROR (errhp,
        OCIStmtExecute (svchp, p_sql, errhp, (ub4) 1, (ub4) 0,
                (OCISnapshot *) NULL, (OCISnapshot *) NULL,
                (ub4) OCI_COMMIT_ON_SUCCESS));
  printf (" Field2 After:\n");
  printf (" size ---> %d\n", sizeof (field2));
  printf (" length ---> %d\n", strlen (field2));
  printf (" value ---> %s\n", field2);
  HANDLE_ERROR (errhp, OCIHandleFree (p_sql, OCI_HTYPE_STMT));
}
void
call_stored_proc2 (OCISvcCtx * svchp, OCIError * errhp, OCIEnv * envhp)
{
  OCIStmt *p_sql;
  OCIBind *p_Bind1 = (OCIBind *) 0;
  OCIBind *p_Bind2 = (OCIBind *) 0;
  OCIBind *p_Bind3 = (OCIBind *) 0;
  char field2[20] = "Entry 3";
  char field3[20];
  sword field1 = 3;
  text *mySql =
    (text *) "Begin StoredProcedureSample2(:MGR, :ENAME, :EOUT); END";
  memset( field2, 0, sizeof(field2) );
  strcpy( field2, "Entry 3" );
  memset( field3, 0, sizeof(field3) );
  printf ("*************************************************\n");
  printf ("Example 2 - Using an OUT Parameter\n");
  printf ("*************************************************\n");
  /*
   * Initialize & Allocate Statement Handle
   */
  HANDLE_ERROR (errhp,
        OCIHandleAlloc (envhp, (dvoid **) & p_sql,
```

```
                OCI_HTYPE_STMT, (size_t) 0, (dvoid **) 0));
  HANDLE_ERROR (errhp,
        OCIStmtPrepare (p_sql, errhp, mySql,
                (ub4) strlen ((char *)mySql), OCI_NTV_SYNTAX,
                OCI_DEFAULT));
  HANDLE_ERROR (errhp,
        OCIBindByPos (p_sql, &p_Bind1, errhp, 1,
                  (dvoid *) & field1, sizeof (sword),
                  SQLT_INT, 0, 0, 0, 0, 0, OCI_DEFAULT));
  HANDLE_ERROR (errhp,
        OCIBindByPos (p_sql, &p_Bind2, errhp, 2,
                  field2, strlen (field2) + 1,
                  SQLT_STR, 0, 0, 0, 0, 0, OCI_DEFAULT));
  HANDLE_ERROR (errhp,
        OCIBindByPos (p_sql, &p_Bind3, errhp, 3,
                  field3, 20,
                  SQLT_STR, 0, 0, 0, 0, 0, OCI_DEFAULT));
  printf (" Field3 Before:\n");
  printf (" size ---> %d\n", sizeof (field3));
  printf (" length ---> %d\n", strlen (field3));
  printf (" value ---> %s\n", field3);
  HANDLE_ERROR (errhp,
        OCIStmtExecute (svchp, p_sql, errhp, (ub4) 1, (ub4) 0,
                (OCISnapshot *) NULL, (OCISnapshot *) NULL,
                (ub4) OCI_COMMIT_ON_SUCCESS));
  printf (" Field3 After:\n");
  printf (" size ---> %d\n", sizeof (field3));
  printf (" length ---> %d\n", strlen (field3));
  printf (" value ---> %s\n", field3);
  HANDLE_ERROR (errhp, OCIHandleFree (p_sql, OCI_HTYPE_STMT));
}
/* drop table(s) required for this example */
void
drop_table (OCISvcCtx * svchp, OCIError * errhp, OCIEnv * envhp)
{
  OCIStmt *stmhp;
  text *statement = (text *)"DROP TABLE OCISPEC";
  ub4 status = OCI_SUCCESS;
  /*
   * Initialize & Allocate Statement Handle
   */
  HANDLE_ERROR (errhp,
        OCIHandleAlloc (envhp, (dvoid **) & stmhp,
                OCI_HTYPE_STMT, (size_t) 0, (dvoid **) 0));
  /*
   * Prepare the drop statement
   */
  HANDLE_ERROR (errhp,
        OCIStmtPrepare (stmhp, errhp,
                statement, strlen ((const char *) statement),
                OCI_NTV_SYNTAX, OCI_DEFAULT));
  /*
   * Execute the drop Statement
   */
  if ((status = OCIStmtExecute (svchp, stmhp, errhp,
                  (ub4) 1, (ub4) 0, NULL, NULL, OCI_DEFAULT)) < OCI_SUCCESS)
    {
      printf ("FAILURE IN DROPING TABLE(S)\n");
      HANDLE_ERROR (errhp, status);
      return;
```

```
      return;
    }
  else
    {
      printf ("*******************************************\n");
      printf ("MileStone : Table(s) Successfully Dropped\n");
      printf ("*******************************************\n");
    }
  HANDLE_ERROR (errhp, OCIHandleFree (stmhp, OCI_HTYPE_STMT));
}
void
drop_stored_procs (OCISvcCtx * svchp, OCIError * errhp, OCIEnv * envhp)
{
  OCIStmt *stmhp;
  OCIStmt *stmhp2;
  OCIStmt *stmhp3;
  text *create_statement = (text *)"DROP PROCEDURE StoredProcedureSample1";
  text *create_statement2 = (text *)"DROP PROCEDURE StoredProcedureSample2";
  text *create_statement3 = (text *)"DROP FUNCTION  f1";
  ub4 status = OCI_SUCCESS;
  OCITransCommit( svchp, errhp, OCI_DEFAULT );
  /*
   * Initialize & Allocate Statement Handles
   */
  HANDLE_ERROR (errhp,
        OCIHandleAlloc (envhp, (dvoid **) & stmhp,
                OCI_HTYPE_STMT, (size_t) 0, (dvoid **) 0));
  HANDLE_ERROR (errhp,
        OCIHandleAlloc (envhp, (dvoid **) & stmhp2,
                OCI_HTYPE_STMT, (size_t) 0, (dvoid **) 0));
  HANDLE_ERROR (errhp,
        OCIHandleAlloc (envhp, (dvoid **) & stmhp3,
                OCI_HTYPE_STMT, (size_t) 0, (dvoid **) 0));
  /*
   * Prepare the Create statements
   */
  HANDLE_ERROR (errhp,
        OCIStmtPrepare (stmhp, errhp,
                create_statement,
                strlen ((const char *) create_statement),
                OCI_NTV_SYNTAX, OCI_DEFAULT));
  HANDLE_ERROR (errhp,
        OCIStmtPrepare (stmhp2, errhp, create_statement2,
                strlen ((const char *) create_statement2),
                OCI_NTV_SYNTAX, OCI_DEFAULT));
  HANDLE_ERROR (errhp,
        OCIStmtPrepare (stmhp3, errhp, create_statement3,
                strlen ((const char *) create_statement3),
                OCI_NTV_SYNTAX, OCI_DEFAULT));
  /*
   * Execute the Create Statement SampleProcedure1
   */
  if ((status = OCIStmtExecute (svchp, stmhp, errhp,
                  (ub4) 1, (ub4) 0, NULL, NULL, OCI_DEFAULT)) < OCI_SUCCESS)
    {
      printf ("FAILURE IN DROPPING PROCEDURE 1\n");
      HANDLE_ERROR (errhp, status);
      return;
    }
  else
```

```
    {
      printf ("*********************************************\n");
      printf ("MileStone : Sample Procedure 1 Successfully dropped\n");
      printf ("*********************************************\n");
    }
  /*
   * Execute the Create Statement Sample Procedure2
   */
  if ((status = OCIStmtExecute (svchp, stmhp2, errhp,
                  (ub4) 1, (ub4) 0, NULL, NULL, OCI_DEFAULT)) < OCI_SUCCESS)
    {
      printf ("FAILURE IN DROPPING PROCEDURE 2\n");
      HANDLE_ERROR (errhp, status);
      return;
    }
  else
    {
      printf ("*********************************************\n");
      printf ("MileStone : Sample Procedure 2 Successfully dropped\n");
      printf ("*********************************************\n");
    }
  /*
   * Execute the Create Statement Sample Procedure3
   */
  if ((status = OCIStmtExecute (svchp, stmhp3, errhp,
                  (ub4) 1, (ub4) 0, NULL, NULL, OCI_DEFAULT)) < OCI_SUCCESS)
    {
      printf ("FAILURE IN DROPPING PROCEDURE 3\n");
      HANDLE_ERROR (errhp, status);
      return;
    }
  else
    {
      printf ("*********************************************\n");
      printf ("MileStone : Sample Procedure 3 Successfully dropped\n");
      printf ("*********************************************\n");
    }
  HANDLE_ERROR (errhp, OCIHandleFree (stmhp, OCI_HTYPE_STMT));
  HANDLE_ERROR (errhp, OCIHandleFree (stmhp2, OCI_HTYPE_STMT));
  HANDLE_ERROR (errhp, OCIHandleFree (stmhp3, OCI_HTYPE_STMT));
}
/* Clean your mess up */
void
cleanup (OCISvcCtx ** svchp, OCIServer ** srvhp, OCISession ** authp,
     OCIError ** errhp, OCIEnv ** envhp)
{
  /*
   * log off
   */
  HANDLE_ERROR (*errhp, OCISessionEnd (*svchp, *errhp, *authp, OCI_DEFAULT));
  printf ("logged off\n");
  /*
   * detach from server
   */
  HANDLE_ERROR (*errhp, OCIServerDetach (*srvhp, *errhp, OCI_DEFAULT));
  printf ("detached form server\n");
  /*
   * free up handles
   */
```

```
  HANDLE_ERROR (*errhp, OCIHandleFree (*authp, OCI_HTYPE_SESSION));
  /* free session handle */
  *authp = 0;
  HANDLE_ERROR (*errhp, OCIHandleFree (*srvhp, OCI_HTYPE_SERVER));
  /* free server handle */
  *srvhp = 0;
  HANDLE_ERROR (*errhp, OCIHandleFree (*svchp, OCI_HTYPE_SVCCTX));
  /* free service context */
  *svchp = 0;
  HANDLE_ERROR (*errhp, OCIHandleFree (*errhp, OCI_HTYPE_ERROR));
  /* free error handle */
  *errhp = 0;
  OCIHandleFree (*envhp, OCI_HTYPE_ENV);
  /* free environment handle */
  *envhp = 0;
  printf ("free'd all handles\n");
}
```

In the preceding code example, you must replace the following parameters with the connection information of your PolarDB instance.

Parameter	Example	Description
`text *username`	`(text *) "postgres"`	The username of the PolarDB instance.
`text *passwd`	`(text *) ""`	The password for the username of the PolarDB instance.
`text *server`	`(text *) "//localhost:5432"`	The endpoint and the port of the PolarDB instance. For more information about how to query the connection information, see Create database accounts in *PolarDB User Guide*

Note For more information about the Oracle native OCI driver, visit OCI: Introduction.

Compile code

- Linux
  i. Modify the Makefile file to dynamically link to the path where the polaroci.so file is located.

  The following content is an example of the Makefile file:

```
# ============================================================================
# Copyright (c) 2004-2012 PolarDB Corporation. All Rights Reserved.
# ============================================================================
# Makefile to build C testcases for OCILib
#
CC=gcc
CFLAGS=-Wall -g -I$(ORACLE_HOME)/ -L $(POLARDBOCI_LIB) -lpolardboci -lpq -liconv
SAMPLES = polardb_demo
all: $(SAMPLES)
%:%.o
    $(CC) $(CFLAGS) -o $@
clean:
    rm -rf $(SAMPLES)
```

- Link ORACLE_HOME to the directory *instantclient_12_1/sdk/include* of the oracle oci header file that is downloaded from the driver directory.
- Link POLARDBOCI_LIB to the directory where the libpolardboci.so, libpq.so, and libiconv.so files are located.

ii. Run the following command to compile the code:

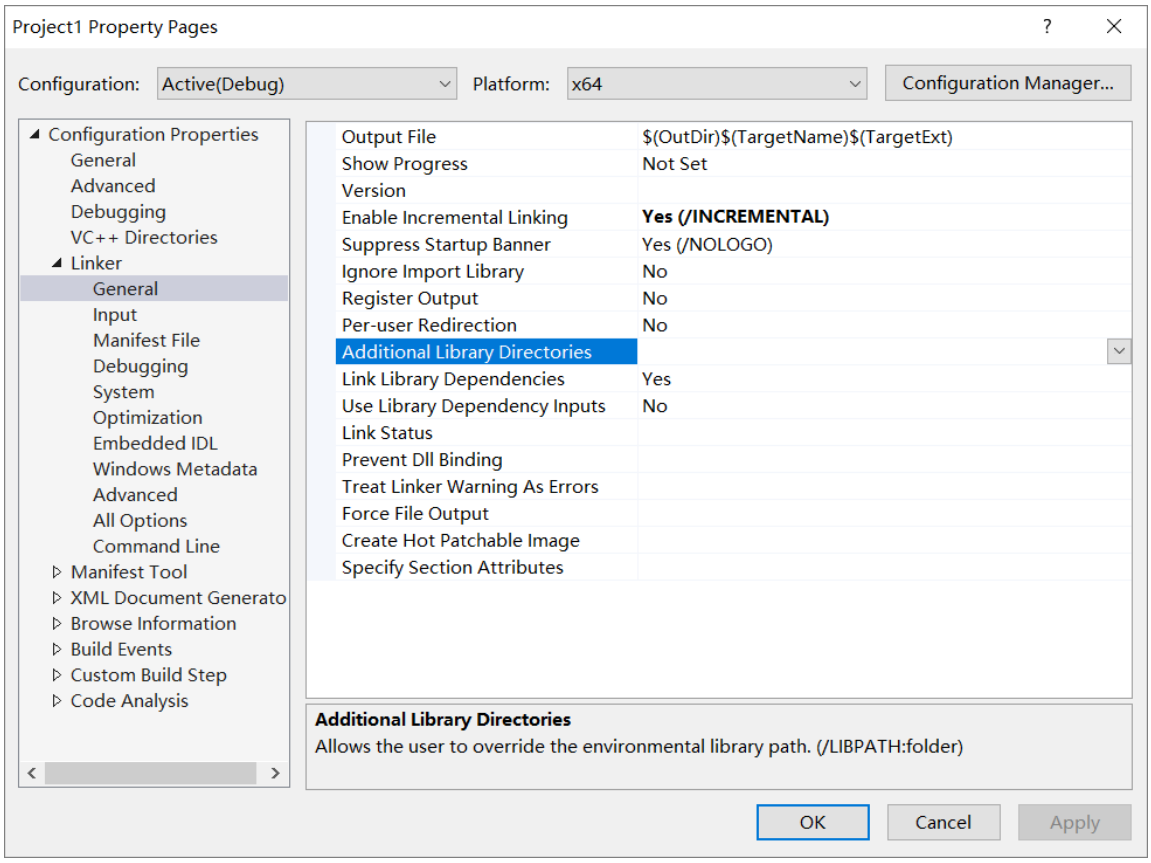
```
make
```

- Windows

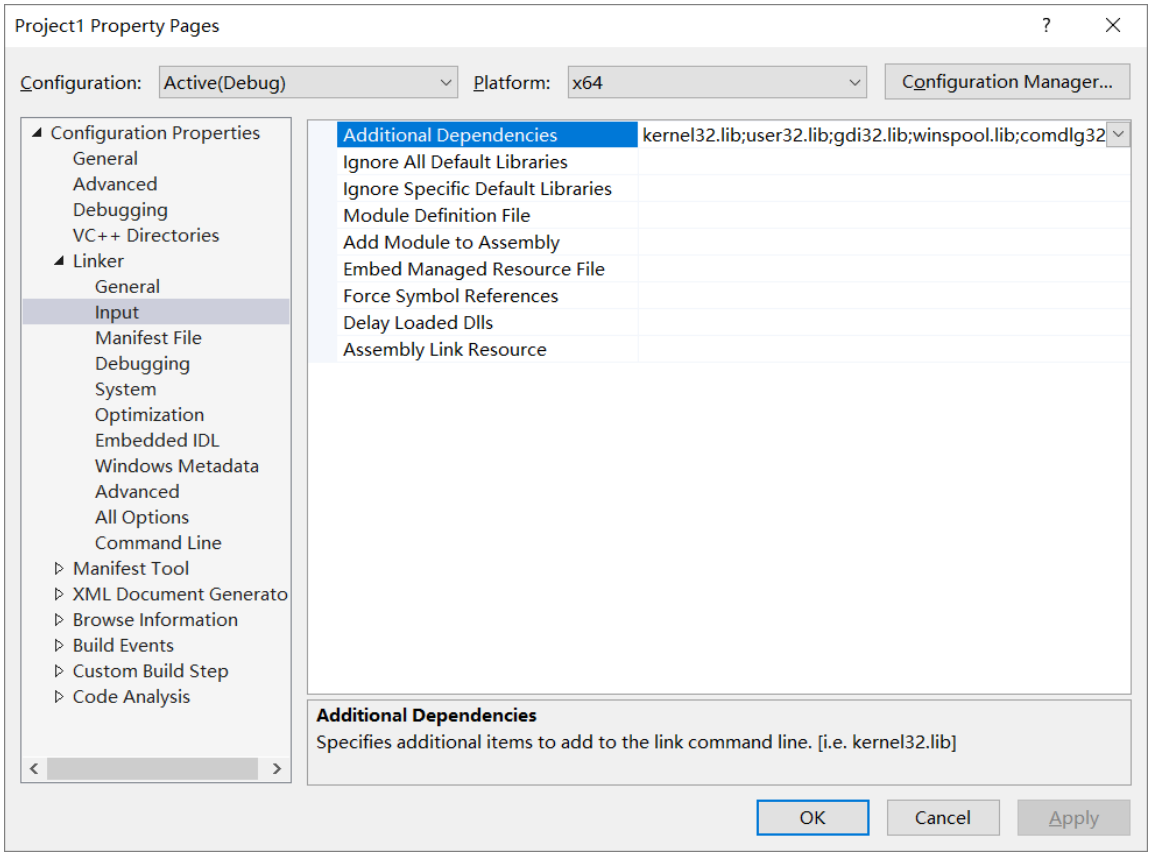
  Visual Studio is used in the example.

  i. Add the path of the oracle oci development package in the driver directory to **C/C++ > General > Attachment Include Directories**.

  ii. Add the paths of polardboci.dll and polardboci.lib in the driver directory to **Linker > General > Additional Library Directories**.



  iii. Enter polardboci.lib in **Linker > Input > Additional Dependencies**.



Examples

The following executable file is obtained after polardb_demo is compiled:

```
*********************************************
Milestone  : Logged on as --> 'parallels'
*********************************************
*********************************************
MileStone : Table(s) Successfully created
*********************************************
*********************************************
MileStone : Data Sucessfully inserted
 & Committed via Transaction
*********************************************
*********************************************
MileStone : Sample Procedure 1 Successfully created
*********************************************
*********************************************
MileStone : Sample Procedure 2 Successfully created
*********************************************
*********************************************
MileStone : Sample Procedure 3 Successfully created
*********************************************
name -> [SMITH]    mgr -> [7886]
hiredate -> [2007-08-02 00:00:00]
name -> [ALLEN]    mgr -> [7110]
hiredate -> [2007-04-02 00:00:00]
name -> [KING]    mgr -> [7221]
hiredate -> [2007-03-02 00:00:00]
*************************************************
Example 1 - Using an IN OUT Parameter
.................................................
```

```
************************************************
 Field2 Before:
 size ---> 20
 length ---> 7
 value ---> Entry 3
 Field2 After:
 size ---> 20
 length ---> 10
 value ---> Successful
************************************************
Example 2 - Using an OUT Parameter
************************************************
 Field3 Before:
 size ---> 20
 length ---> 0
 value --->
 Field3 After:
 size ---> 20
 length ---> 10
 value ---> Successful
******************************************
MileStone : Table(s) Successfully Dropped
******************************************
******************************************
MileStone : Sample Procedure 1 Successfully dropped
******************************************
******************************************
MileStone : Sample Procedure 2 Successfully dropped
******************************************
******************************************
MileStone : Sample Procedure 3 Successfully dropped
******************************************
logged off
detached form server
free'd all handles
```

11.1.15.6. RDS PolarDB PHP

This topic describes how to connect a PHP client to a PolarDB instance.

Prerequisites

An account is created for a PolarDB instance. For more information, see Create an account.

Prepare the environment in Windows

1. Download and install WampServer. For more information, visit WampServer official website.
2. Start the PostgreSQL plug-in.
   i. Modify the `php.ini` file.

ii. Remove semicolons (`;`) from the following code.

Before you remove semicolons:

```
;extension=php_pgsql.dll
;extension=php_pdo_pgsql.dll
```

After you remove semicolons:

```
extension=php_pgsql.dll
extension=php_pdo_pgsql.dll
```

3. Copy the `libpq.dll` file from the *C:\wamp\bin\php\php5.3.5* directory to the *C:\windows\system32* directory. Note: php5.3.5 is used in this example, and the actual directory varies based on your client version.
4. Restart the Apache service.

Prepare the environment in Linux

1. Install the `php-pgsql.x86_64` driver.

```
sudo yum install php-pgsql.x86_64
```

2. Modify the `php.ini` file.

```
vim /etc/php.ini
```

3. Add the following content to the `php.ini` file.

```
extension=php_pgsql.so
```

Connect to PolarDB

After you prepare the environment in Windows or Linux, you can run a PHP script to connect to the PolarDB instance.

The following sample code shows how to use PHP to connect to the PolarDB instance.

```
<?php
 $host    = "host=xxxx";
 $port    = "port=xxxx";
 $dbname   = "dbname=xxxx";
 $credentials = "user=xxxx password=xxxxx";
 $db = pg_connect( "$host $port $dbname $credentials" );
 if(!$db){
  echo "Error : Unable to open database\n";
 } else {
  echo "Opened database successfully\n";
 }
 $sql =<<<EOF
  select * from pg_roles;
EOF;
 $ret = pg_query($db, $sql);
 if(!$ret){
  echo pg_last_error($db);
 } else {
  echo "Records created successfully\n";
 }
 $results = pg_fetch_all($ret);
 print_r($results);
 pg_close($db);
?>
```

In the preceding sample code, the connection information of PolarDB consists of parameters, such as `host`, `port`, `dbname`, and `credentials`, as shown in the following table.

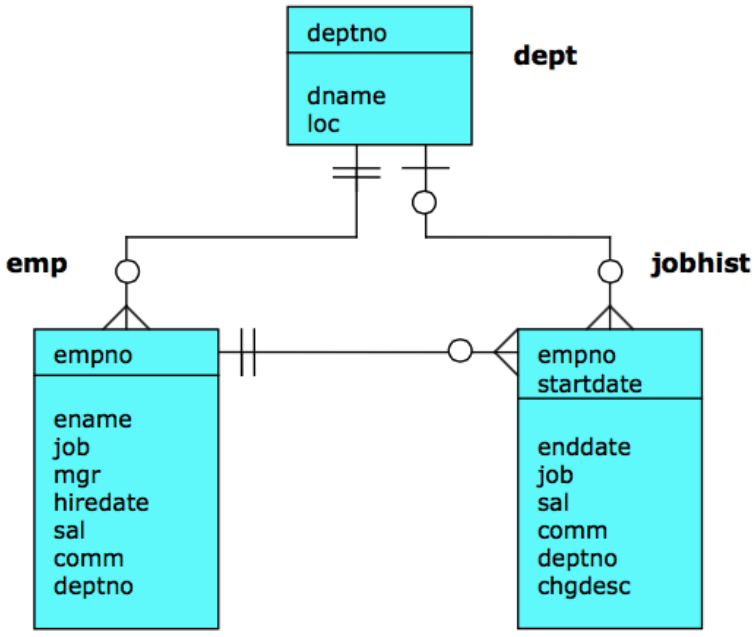
Parameter	Example	Description
host	`"host=xxxxxx"`	The endpoint of the PolarDB instance. For more information about how to view the endpoint, see View or apply for an endpoint in *PolarDB User Guide*.
port	`"port=1521"`	The port of the PolarDB instance.
dbname	`"dbname=xxxx"`	The name of the database to be connected.
credentials	`"user=xxx password=xxxx"`	The username and password used to log on to the PolarDB instance.

For more information about PHP APIs, visit PHP documentation.

11.1.16. Compatibility for Oracle

In this topic, examples are used to help you understand terms and concepts in PolarDB. You can efficiently develop PolarDB databases and migrate data from Oracle to PolarDB databases.

The operations described in this topic are built on a data model that displays basic PolarDB operations, such as database creation, table creation, and user management. The following figure shows the data model:



To simulate an environment similar to Oracle, the following example creates a database named orcl_polardb, a role named scott in the orcl_polardb database, and a schema named scott.

Use psql to connect to a database

```
psql -h polardbaddress.polardb.rds.aliyuncs.com -p 3433 -U myuser -d template1
  Enter the password of the myuser user:
  psql.bin (9.4.1.3, server 9.3.5.14)
  Enter help to obtain help information.
  template1=>
```

CREATE DATABASE

```
template1=> CREATE DATABASE orcl_polardb;
  CREATE DATABASE
  template1=> \c orcl_polardb
  psql.bin (9.4.1.3, server 9.3.5.14)
```

CREATE ROLE

```
 orcl_polardb=> CREATE ROLE scott LOGIN PASSWORD 'scott123';
  CREATE ROLE
```

CREATE SCHEMA

```
  orcl_polardb=> CREATE SCHEMA scott;
  CREATE SCHEMA
  orcl_polardb=> GRANT scott TO myuser;
  GRANT ROLE
  orcl_polardb=> ALTER SCHEMA scott OWNER TO scott;
  ALTER SCHEMA
  orcl_polardb=> REVOKE scott FROM myuser;
  REVOKE ROLE
```

Note

- If you have not granted the permissions owned by the scott role to the myuser user when you execute the `ALTER SCHEMA scott OWNER TO scott` statement, the following permission error is displayed: `ERROR:must be member of role "scott"` .
- For security reasons, revoke the scott permissions from the myuser user after the statement is executed. This improves security.

Connect to the orcl_polardb database

Note The following operations must be performed by using the scott account. Otherwise, the created tables and objects do not belong to scott and permission errors may occur.

```
[root@localhost bin]# ./psql -h polardbaddress.polardb.rds.aliyuncs.com -p 3433 -U scott -d orcl_pola
rdb
Enter the password of the user scott:
psql.bin (9.4.1.3, server 9.3.5.14)
Enter help to obtain help information.
orcl_polardb=>
```

CREATE TABLE

```
CREATE TABLE dept (
   deptno          NUMBER(2) NOT NULL CONSTRAINT dept_pk PRIMARY KEY,
   dname           VARCHAR2(14) CONSTRAINT dept_dname_uq UNIQUE,
   lock            VARCHAR2(13)
);
CREATE TABLE emp (
  empno           NUMBER(4) NOT NULL CONSTRAINT emp_pk PRIMARY KEY,
  ename           VARCHAR2(10),
  job             VARCHAR2(9),
  mgr             NUMBER(4),
  hiredate        DATE,
  sal             NUMBER(7,2) CONSTRAINT emp_sal_ck CHECK (sal > 0),
  comm            NUMBER(7,2),
  deptno          NUMBER(2) CONSTRAINT emp_ref_dept_fk
                  REFERENCES dept(deptno)
);
CREATE TABLE jobhist (
  empno           NUMBER(4) NOT NULL,
  startdate       DATE NOT NULL,
  enddate         DATE,
  job             VARCHAR2(9),
  sal             NUMBER(7,2),
  comm            NUMBER(7,2),
  deptno          NUMBER(2),
  chgdesc         VARCHAR2(80),
  CONSTRAINT jobhist_pk PRIMARY KEY (empno, startdate),
  CONSTRAINT jobhist_ref_emp_fk FOREIGN KEY (empno)
      REFERENCES emp(empno) ON DELETE CASCADE,
  CONSTRAINT jobhist_ref_dept_fk FOREIGN KEY (deptno)
      REFERENCES dept (deptno) ON DELETE SET NULL,
  CONSTRAINT jobhist_date_chk CHECK (startdate <= enddate)
);
```

CREATE OR REPLACE VIEW

```
CREATE OR REPLACE VIEW salesemp AS
   SELECT empno, ename, hiredate, sal, comm FROM emp WHERE job = 'SALESMAN';
```

CREATE SEQUENCE

```
CREATE SEQUENCE next_empno START WITH 8000 INCREMENT BY 1;
```

INSERT INTO

```
INSERT INTO dept VALUES (10,'ACCOUNTING','NEW YORK');
INSERT INTO dept VALUES (20,'RESEARCH','DALLAS');
INSERT INTO dept VALUES (30,'SALES','CHICAGO');
INSERT INTO dept VALUES (40,'OPERATIONS','BOSTON');
INSERT INTO emp VALUES (7369,'SMITH','CLERK',7902,'17-DEC-80',800,NULL,20);
INSERT INTO emp VALUES (7499,'ALLEN','SALESMAN',7698,'20-FEB-81',1600,300,30);
INSERT INTO emp VALUES (7521,'WARD','SALESMAN',7698,'22-FEB-81',1250,500,30);
INSERT INTO emp VALUES (7566,'JONES','MANAGER',7839,'02-APR-81',2975,NULL,20);
INSERT INTO emp VALUES (7654,'MARTIN','SALESMAN',7698,'28-SEP-81',1250,1400,30);
INSERT INTO emp VALUES (7698,'BLAKE','MANAGER',7839,'01-MAY-81',2850,NULL,30);
INSERT INTO emp VALUES (7782,'CLARK','MANAGER',7839,'09-JUN-81',2450,NULL,10);
INSERT INTO emp VALUES (7788,'SCOTT','ANALYST',7566,'19-APR-87',3000,NULL,20);
INSERT INTO emp VALUES (7839,'KING','PRESIDENT',NULL,'17-NOV-81',5000,NULL,10);
INSERT INTO emp VALUES (7844,'TURNER','SALESMAN',7698,'08-SEP-81',1500,0,30);
INSERT INTO emp VALUES (7876,'ADAMS','CLERK',7788,'23-MAY-87',1100,NULL,20);
INSERT INTO emp VALUES (7900,'JAMES','CLERK',7698,'03-DEC-81',950,NULL,30);
INSERT INTO emp VALUES (7902,'FORD','ANALYST',7566,'03-DEC-81',3000,NULL,20);
INSERT INTO emp VALUES (7934,'MILLER','CLERK',7782,'23-JAN-82',1300,NULL,10);
INSERT INTO jobhist VALUES (7369,'17-DEC-80',NULL,'CLERK',800,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7499,'20-FEB-81',NULL,'SALESMAN',1600,300,30,'New Hire');
INSERT INTO jobhist VALUES (7521,'22-FEB-81',NULL,'SALESMAN',1250,500,30,'New Hire');
INSERT INTO jobhist VALUES (7566,'02-APR-81',NULL,'MANAGER',2975,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7654,'28-SEP-81',NULL,'SALESMAN',1250,1400,30,'New Hire');
INSERT INTO jobhist VALUES (7698,'01-MAY-81',NULL,'MANAGER',2850,NULL,30,'New Hire');
INSERT INTO jobhist VALUES (7782,'09-JUN-81',NULL,'MANAGER',2450,NULL,10,'New Hire');
INSERT INTO jobhist VALUES (7788,'19-APR-87','12-APR-88','CLERK',1000,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7788,'13-APR-88','04-MAY-89','CLERK',1040,NULL,20,'Raise');
INSERT INTO jobhist VALUES (7788,'05-MAY-90',NULL,'ANALYST',3000,NULL,20,'Promoted to Analyst');
INSERT INTO jobhist VALUES (7839,'17-NOV-81',NULL,'PRESIDENT',5000,NULL,10,'New Hire');
INSERT INTO jobhist VALUES (7844,'08-SEP-81',NULL,'SALESMAN',1500,0,30,'New Hire');
INSERT INTO jobhist VALUES (7876,'23-MAY-87',NULL,'CLERK',1100,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7900,'03-DEC-81','14-JAN-83','CLERK',950,NULL,10,'New Hire');
INSERT INTO jobhist VALUES (7900,'15-JAN-83',NULL,'CLERK',950,NULL,30,'Changed to Dept 30');
INSERT INTO jobhist VALUES (7902,'03-DEC-81',NULL,'ANALYST',3000,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7934,'23-JAN-82',NULL,'CLERK',1300,NULL,10,'New Hire');
```

ANALYZE

```
ANALYZE dept;
ANALYZE emp;
ANALYZE jobhist;
```

CREATE PROCEDURE

```
CREATE OR REPLACE PROCEDURE list_emp
IS
  v_empno         NUMBER(4);
  v_ename         VARCHAR2(10);
  CURSOR emp_cur IS
      SELECT empno, ename FROM emp ORDER BY empno;
BEGIN
  OPEN emp_cur;
  DBMS_OUTPUT.PUT_LINE('EMPNO    ENAME');
  DBMS_OUTPUT.PUT_LINE('-----    -------');
  LOOP
      FETCH emp_cur INTO v_empno, v_ename;
      EXIT WHEN emp_cur%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE(v_empno || '    ' || v_ename);
```

```
      DBMS_OUTPUT.PUT_LINE(v_empno || ' ' || v_ename);
  END LOOP;
  CLOSE emp_cur;
END;
--
--  Procedure that selects an employee row given the employee
--  number and displays certain columns.
--
CREATE OR REPLACE PROCEDURE select_emp (
  p_empno         IN  NUMBER
)
IS
  v_ename         emp.ename%TYPE;
  v_hiredate      emp.hiredate%TYPE;
  v_sal           emp.sal%TYPE;
  v_comm          emp.comm%TYPE;
  v_dname         dept.dname%TYPE;
  v_disp_date     VARCHAR2(10);
BEGIN
  SELECT ename, hiredate, sal, NVL(comm, 0), dname
      INTO v_ename, v_hiredate, v_sal, v_comm, v_dname
      FROM emp e, dept d
      WHERE empno = p_empno
        AND e.deptno = d.deptno;
  v_disp_date := TO_CHAR(v_hiredate, 'MM/DD/YYYY');
  DBMS_OUTPUT.PUT_LINE('Number    : ' || p_empno);
  DBMS_OUTPUT.PUT_LINE('Name      : ' || v_ename);
  DBMS_OUTPUT.PUT_LINE('Hire Date : ' || v_disp_date);
  DBMS_OUTPUT.PUT_LINE('Salary    : ' || v_sal);
  DBMS_OUTPUT.PUT_LINE('Commission: ' || v_comm);
  DBMS_OUTPUT.PUT_LINE('Department: ' || v_dname);
EXCEPTION
  WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('Employee ' || p_empno || ' not found');
  WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('The following is SQLERRM:');
      DBMS_OUTPUT.PUT_LINE(SQLERRM);
      DBMS_OUTPUT.PUT_LINE('The following is SQLCODE:');
      DBMS_OUTPUT.PUT_LINE(SQLCODE);
END;
--
--  Procedure that queries the 'emp' table based on
--  department number and employee number or name.  Returns
--  employee number and name as IN OUT parameters and job,
--  hire date, and salary as OUT parameters.
--
CREATE OR REPLACE PROCEDURE emp_query (
  p_deptno        IN     NUMBER,
  p_empno         IN OUT NUMBER,
  p_ename         IN OUT VARCHAR2,
  p_job           OUT    VARCHAR2,
  p_hiredate      OUT    DATE
  p_sal           OUT    NUMBER
)
IS
BEGIN
  SELECT empno, ename, job, hiredate, sal
      INTO p_empno, p_ename, p_job, p_hiredate, p_sal
      FROM emp
      WHERE deptno = p_deptno
```

```
        AND (empno = p_empno
         OR  ename = UPPER(p_ename));
END;
--
--  Procedure to call 'emp_query_caller' with IN and IN OUT
--  parameters.  Displays the results received from IN OUT and
--  OUT parameters.
--
CREATE OR REPLACE PROCEDURE emp_query_caller
IS
  v_deptno        NUMBER(2);
  v_empno         NUMBER(4);
  v_ename         VARCHAR2(10);
  v_job           VARCHAR2(9);
  v_hiredate      DATE;
  v_sal           NUMBER;
BEGIN
  v_deptno := 30;
  v_empno  := 0;
  v_ename  := 'Martin';
  emp_query(v_deptno, v_empno, v_ename, v_job, v_hiredate, v_sal);
  DBMS_OUTPUT.PUT_LINE('Department : ' || v_deptno);
  DBMS_OUTPUT.PUT_LINE('Employee No: ' || v_empno);
  DBMS_OUTPUT.PUT_LINE('Name       : ' || v_ename);
  DBMS_OUTPUT.PUT_LINE('Job        : ' || v_job);
  DBMS_OUTPUT.PUT_LINE('Hire Date  : ' || v_hiredate);
  DBMS_OUTPUT.PUT_LINE('Salary     : ' || v_sal);
EXCEPTION
  WHEN TOO_MANY_ROWS THEN
      DBMS_OUTPUT.PUT_LINE('More than one employee was selected');
  WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('No employees were selected');
END;
```

CREATE FUNCTION

```
CREATE OR REPLACE FUNCTION emp_comp (
  p_sal           NUMBER,
  p_comm          NUMBER
) RETURN NUMBER
IS
BEGIN
  RETURN (p_sal + NVL(p_comm, 0)) * 24;
END;
--
--  Function that gets the next number from sequence, 'next_empno',
--  and ensures it is not already in use as an employee number.
--
CREATE OR REPLACE FUNCTION new_empno RETURN NUMBER
IS
  v_cnt           INTEGER := 1;
  v_new_empno     NUMBER;
BEGIN
  WHILE v_cnt > 0 LOOP
      SELECT next_empno.nextval INTO v_new_empno FROM dual;
      SELECT COUNT(*) INTO v_cnt FROM emp WHERE empno = v_new_empno;
  END LOOP;
  RETURN v_new_empno;
END;
```

```
END;
--
--  EDB-SPL function that adds a new clerk to table 'emp'.  This function
--  uses package 'emp_admin'.
--
CREATE OR REPLACE FUNCTION hire_clerk (
  p_ename         VARCHAR2,
  p_deptno        NUMBER
) RETURN NUMBER
IS
  v_empno         NUMBER(4);
  v_ename         VARCHAR2(10);
  v_job           VARCHAR2(9);
  v_mgr           NUMBER(4);
  v_hiredate      DATE;
  v_sal           NUMBER(7,2);
  v_comm          NUMBER(7,2);
  v_deptno        NUMBER(2);
BEGIN
  v_empno := new_empno;
  INSERT INTO emp VALUES (v_empno, p_ename, 'CLERK', 7782,
      TRUNC(SYSDATE), 950.00, NULL, p_deptno);
  SELECT empno, ename, job, mgr, hiredate, sal, comm, deptno INTO
      v_empno, v_ename, v_job, v_mgr, v_hiredate, v_sal, v_comm, v_deptno
      FROM emp WHERE empno = v_empno;
  DBMS_OUTPUT.PUT_LINE('Department : ' || v_deptno);
  DBMS_OUTPUT.PUT_LINE('Employee No: ' || v_empno);
  DBMS_OUTPUT.PUT_LINE('Name       : ' || v_ename);
  DBMS_OUTPUT.PUT_LINE('Job        : ' || v_job);
  DBMS_OUTPUT.PUT_LINE('Manager    : ' || v_mgr);
  DBMS_OUTPUT.PUT_LINE('Hire Date  : ' || v_hiredate);
  DBMS_OUTPUT.PUT_LINE('Salary     : ' || v_sal);
  DBMS_OUTPUT.PUT_LINE('Commission : ' || v_comm);
  RETURN v_empno;
EXCEPTION
  WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('The following is SQLERRM:');
      DBMS_OUTPUT.PUT_LINE(SQLERRM);
      DBMS_OUTPUT.PUT_LINE('The following is SQLCODE:');
      DBMS_OUTPUT.PUT_LINE(SQLCODE);
      RETURN -1;
END;
--
--  PostgreSQL PL/pgSQL function that adds a new salesman
--  to table 'emp'.
--
CREATE OR REPLACE FUNCTION hire_salesman (
  p_ename         VARCHAR,
  p_sal           NUMERIC,
  p_comm          NUMERIC
) RETURNS NUMERIC
AS $$
DECLARE
  v_empno         NUMERIC(4);
  v_ename         VARCHAR(10);
  v_job           VARCHAR(9);
  v_mgr           NUMERIC(4);
  v_hiredate      DATE;
  v_sal           NUMERIC(7,2);
  v_comm          NUMERIC(7,2);
```

```
  v_deptno        NUMERIC(2);
BEGIN
  v_empno := new_empno();
  INSERT INTO emp VALUES (v_empno, p_ename, 'SALESMAN', 7698,
      CURRENT_DATE, p_sal, p_comm, 30);
  SELECT INTO
      v_empno, v_ename, v_job, v_mgr, v_hiredate, v_sal, v_comm, v_deptno
      empno, ename, job, mgr, hiredate, sal, comm, deptno
      FROM emp WHERE empno = v_empno;
  RAISE INFO 'Department : %', v_deptno;
  RAISE INFO 'Employee No: %', v_empno;
  RAISE INFO 'Name       : %', v_ename;
  RAISE INFO 'Job        : %', v_job;
  RAISE INFO 'Manager    : %', v_mgr;
  RAISE INFO 'Hire Date  : %', v_hiredate;
  RAISE INFO 'Salary     : %', v_sal;
  RAISE INFO 'Commission : %', v_comm;
  RETURN v_empno;
EXCEPTION
  WHEN OTHERS THEN
      RAISE INFO 'The following is SQLERRM:';
      RAISE INFO '%', SQLERRM;
      RAISE INFO 'The following is SQLSTATE:';
      RAISE INFO '%', SQLSTATE;
      RETURN -1;
END;
```

CREATE RULE

```
CREATE OR REPLACE RULE salesemp_i AS ON INSERT TO salesemp
DO INSTEAD
  INSERT INTO emp VALUES (NEW.empno, NEW.ename, 'SALESMAN', 7698,
      NEW.hiredate, NEW.sal, NEW.comm, 30);
CREATE OR REPLACE RULE salesemp_u AS ON UPDATE TO salesemp
DO INSTEAD
  UPDATE emp SET empno    = NEW.empno,
                 ename    = NEW.ename,
                 hiredate = NEW.hiredate,
                 sal      = NEW.sal,
                 comm     = NEW.comm
      WHERE empno = OLD.empno;
CREATE OR REPLACE RULE salesemp_d AS ON DELETE TO salesemp
DO INSTEAD
DELETE FROM emp WHERE empno = OLD.empno;
```

CREATE TRIGGER

```
CREATE OR REPLACE TRIGGER user_audit_trig
  AFTER INSERT OR UPDATE OR DELETE ON emp
DECLARE
  v_action        VARCHAR2(24);
BEGIN
  IF INSERTING THEN
      v_action := ' added employee(s) on ';
  ELSIF UPDATING THEN
      v_action := ' updated employee(s) on ';
  ELSIF DELETING THEN
      v_action := ' deleted employee(s) on ';
  END IF;
  DBMS_OUTPUT.PUT_LINE('User ' || USER || v_action || TO_CHAR(SYSDATE,'YYYY-MM-DD'));
END;
CREATE OR REPLACE TRIGGER emp_sal_trig
  BEFORE DELETE OR INSERT OR UPDATE ON emp
  FOR EACH ROW
DECLARE
  sal_diff       NUMBER;
BEGIN
  IF INSERTING THEN
      DBMS_OUTPUT.PUT_LINE('Inserting employee ' || :NEW.empno);
      DBMS_OUTPUT.PUT_LINE('..New salary: ' || :NEW.sal);
  END IF;
  IF UPDATING THEN
      sal_diff := :NEW.sal - :OLD.sal;
      DBMS_OUTPUT.PUT_LINE('Updating employee ' || :OLD.empno);
      DBMS_OUTPUT.PUT_LINE('..Old salary: ' || :OLD.sal);
      DBMS_OUTPUT.PUT_LINE('..New salary: ' || :NEW.sal);
      DBMS_OUTPUT.PUT_LINE('..Raise     : ' || sal_diff);
  END IF;
  IF DELETING THEN
      DBMS_OUTPUT.PUT_LINE('Deleting employee ' || :OLD.empno);
      DBMS_OUTPUT.PUT_LINE('..Old salary: ' || :OLD.sal);
  END IF;
END;
```

CREATE PACKAGE

```
CREATE OR REPLACE PACKAGE emp_admin
IS
  FUNCTION get_dept_name (
      p_deptno        NUMBER
  ) RETURN VARCHAR2;
  FUNCTION update_emp_sal (
      p_empno         NUMBER,
      p_raise         NUMBER
  ) RETURN NUMBER;
  PROCEDURE hire_emp (
      p_empno         NUMBER,
      p_ename         VARCHAR2,
      p_job           VARCHAR2,
      p_sal           NUMBER,
      p_hiredate      DATE,
      p_comm          NUMBER,
      p_mgr           NUMBER,
      p_deptno        NUMBER
  );
  PROCEDURE fire_emp (
      p_empno         NUMBER
  );
END emp_admin;
```

CREATE PACKAGE BODY

```
--
--  Package body for the 'emp_admin' package.
--
CREATE OR REPLACE PACKAGE BODY emp_admin
IS
  --
  --  Function that queries the 'dept' table based on the department
  --  number and returns the corresponding department name.
  --
  FUNCTION get_dept_name (
      p_deptno        IN NUMBER
  ) RETURN VARCHAR2
  IS
      v_dname         VARCHAR2(14);
  BEGIN
      SELECT dname INTO v_dname FROM dept WHERE deptno = p_deptno;
      RETURN v_dname;
  EXCEPTION
      WHEN NO_DATA_FOUND THEN
          DBMS_OUTPUT.PUT_LINE('Invalid department number ' || p_deptno);
          RETURN '';
  END;
  --
  --  Function that updates an employee's salary based on the
  --  employee number and salary increment/decrement passed
  --  as IN parameters.  Upon successful completion the function
  --  returns the new updated salary.
  --
  FUNCTION update_emp_sal (
      p_empno         IN NUMBER,
      p_raise         IN NUMBER
  ) RETURN NUMBER
```

```
    IS
        v_sal           NUMBER := 0;
    BEGIN
        SELECT sal INTO v_sal FROM emp WHERE empno = p_empno;
        v_sal := v_sal + p_raise;
        UPDATE emp SET sal = v_sal WHERE empno = p_empno;
        RETURN v_sal;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Employee ' || p_empno || ' not found');
            RETURN -1;
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('The following is SQLERRM:');
            DBMS_OUTPUT.PUT_LINE(SQLERRM);
            DBMS_OUTPUT.PUT_LINE('The following is SQLCODE:');
            DBMS_OUTPUT.PUT_LINE(SQLCODE);
            RETURN -1;
    END;
    --
    --  Procedure that inserts a new employee record into the 'emp' table.
    --
    PROCEDURE hire_emp (
        p_empno         NUMBER,
        p_ename         VARCHAR2,
        p_job           VARCHAR2,
        p_sal           NUMBER,
        p_hiredate      DATE,
        p_comm          NUMBER,
        p_mgr           NUMBER,
        p_deptno        NUMBER
    )
    AS
    BEGIN
        INSERT INTO emp(empno, ename, job, sal, hiredate, comm, mgr, deptno)
            VALUES(p_empno, p_ename, p_job, p_sal,
                   p_hiredate, p_comm, p_mgr, p_deptno);
    END;
    --
    --  Procedure that deletes an employee record from the 'emp' table based
    --  on the employee number.
    --
    PROCEDURE fire_emp (
        p_empno         NUMBER
    )
    AS
    BEGIN
        DELETE FROM emp WHERE empno = p_empno;
    END;
END;
```

11.1.17. Management functions

You cannot use superuser accounts to manage database objects of PolarDB. Therefore, PolarDB provides management functions to help you use various PolarDB features. This topic describes how to use the management functions.

Rules of management functions

You must use the root account of PolarDB to run management functions. The root account is a management account specified when an instance is created and has the CREATEDB, CREATEROLE, and LOGIN permissions.

- rds_manage_extension

  This function allows you to manage plug-ins. You can use this function to create and delete plug-ins supported by PolarDB.

```
rds_manage_extension(operation text, pname text, schema text default NULL,logging bool default fal
se)
  operation: The parameter value is create or drop.
  pname: the name of the supported plug-in.
  schema: the target plug-in mode.
  logging: the log information when the plug-in is created.
  The following plug-ins are supported:
  pg_stat_statements
  btree_gin
  btree_gist
  chkpass
  citext
  cube
  dblink
  dict_int
  earthdistance
  hstore
  intagg
  intarray
  isn
  ltree
  pgcrypto
  pgrowlocks
  pg_prewarm
  pg_trgm
  postgres_fdw
  sslinfo
  tablefunc
  tsearch2
  unaccent
  postgis
  postgis_topology
  fuzzystrmatch
  postgis_tiger_geocoder
  plperl
  pltcl
  plv8
  "uuid-ossp"
  plpgsql
  oss_fdw
  Examples:
  1. Create the dblink plug-in.
      select rds_manage_extension('create','dblink');
  2. Delete the dblink plug-in.
      select rds_manage_extension('drop','dblink');
```

- rds_pg_stat_activity()

  This function returns the information of all connected sessions, which is similar to the pg_stat_activity view.

- rds_pg_stat_statements()

This function encapsulates the pg_stat_statements view. You can use this function to view slow SQL statements within your permissions.

- Performance analysis functions

  The functions allow you to analyze the real-time performance of PolarDB instances and are similar to Automatic Workload Repository (AWR) of Oracle.

```
  1. rds_truncsnap()
  Description: The function is used to delete stored snapshots.
  2 rds_get_snaps()
  Description: The function is used to obtain the information of stored snapshots.
  3 rds_snap()
  Description: The function is used to generate a real-time snapshot.
  4 rds_report(beginsnap bigint, endsnap bigint)
  Description: The function is used to generate a performance analysis report based on snapshots t
hat you specify.
  Example: The following statements generates a performance analysis report based on snapshots.
  SELECT * FROM rds_truncsnap(); // Delete stored snapshots.
  SELECT * from rds_snap(); // Generate a snapshot.
  SELECT * from rds_snap(); // Generate a snapshot.
  SELECT * from rds_snap(); // Generate a snapshot.
  SELECT * FROM rds_get_snaps(); // Obtain the IDs of the generated snapshots, which are 1, 2, and
3.
  SELECT * FROM edbreport(1, 3); // Generate a performance analysis report based on the snapshot 1
and snapshot 3.
```

- Session termination functions

```
 rds_pg_terminate_backend(upid int)
  rds_pg_cancel_backend(upid int)
  The functions are similar to the native pg_terminate_backend and pg_cancel_backend functions. Th
e functions provided by PolarDB cannot manage sessions established by the superuser account.
  Example: The following statement terminates session 123456.
  select rds_pg_cancel_backend(123456);
```

- VPD functions

  Virtual Private Database (VPD) is an encapsulation that is compatible with the DBMS_RLS package and uses the same parameters as DBMS_RLS uses.

```
1. rds_drop_policy is similar to DBMS_RLS.DROP_POLICY.
  2. rds_enable_policy is similar to DBMS_RLS.ENABLE_POLICY.
  3. rds_add_policy is similar to DBMS_RLS.ADD_POLICY.
```

For more information, see Reference of VPD.

12.ApsaraDB RDS for PostgreSQL

12.1. User Guide(RDS PostgreSQL)

12.1.1. What is ApsaraDB RDS?

ApsaraDB RDS is a stable, reliable, and scalable online database service. Based on the distributed file system and high-performance storage, ApsaraDB RDS allows you to perform database operations and maintenance with its set of solutions for disaster recovery, backup, restoration, monitoring, and migration.

ApsaraDB RDS supports four database engines, which are MySQL, SQL Server, PolarDB, and PostgreSQL. You can create database instances based on these database engines to meet your business requirements. This topic describes the PostgreSQL engine.

ApsaraDB RDS for PostgreSQL

ApsaraDB RDS for PostgreSQL is the most advanced open source database. It is fully compatible with SQL and supports a diverse range of data formats such as JSON, IP, and geometric data. In addition to features such as transactions, subqueries, multi-version concurrency control (MVCC), and data integrity check, ApsaraDB RDS for PostgreSQL integrates a series of features including high availability, backup, and restoration to ease operation and maintenance loads.

12.1.2. Limits on ApsaraDB RDS for PostgreSQL

Before you use ApsaraDB RDS for PostgreSQL, you must understand its limits and take the necessary precautions.

The following table describes the limits on ApsaraDB RDS for PostgreSQL.

Operation	Limit
Root permissions of databases	Superuser permissions are not provided.
Database replication	ApsaraDB RDS for PostgreSQL provides a primary/secondary replication architecture except in the Basic Edition. The secondary instances in the architecture are hidden and cannot be accessed by your applications.
Instance restart	Instances must be restarted by using the ApsaraDB RDS console or API operations.

12.1.3. Log on to the ApsaraDB RDS console

This topic describes how to log on to the ApsaraDB RDS console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

> **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
>
> - Uppercase or lowercase letters
> - Digits
> - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products** > Database Services > **ApsaraDB RDS**.

12.1.4. Quick Start

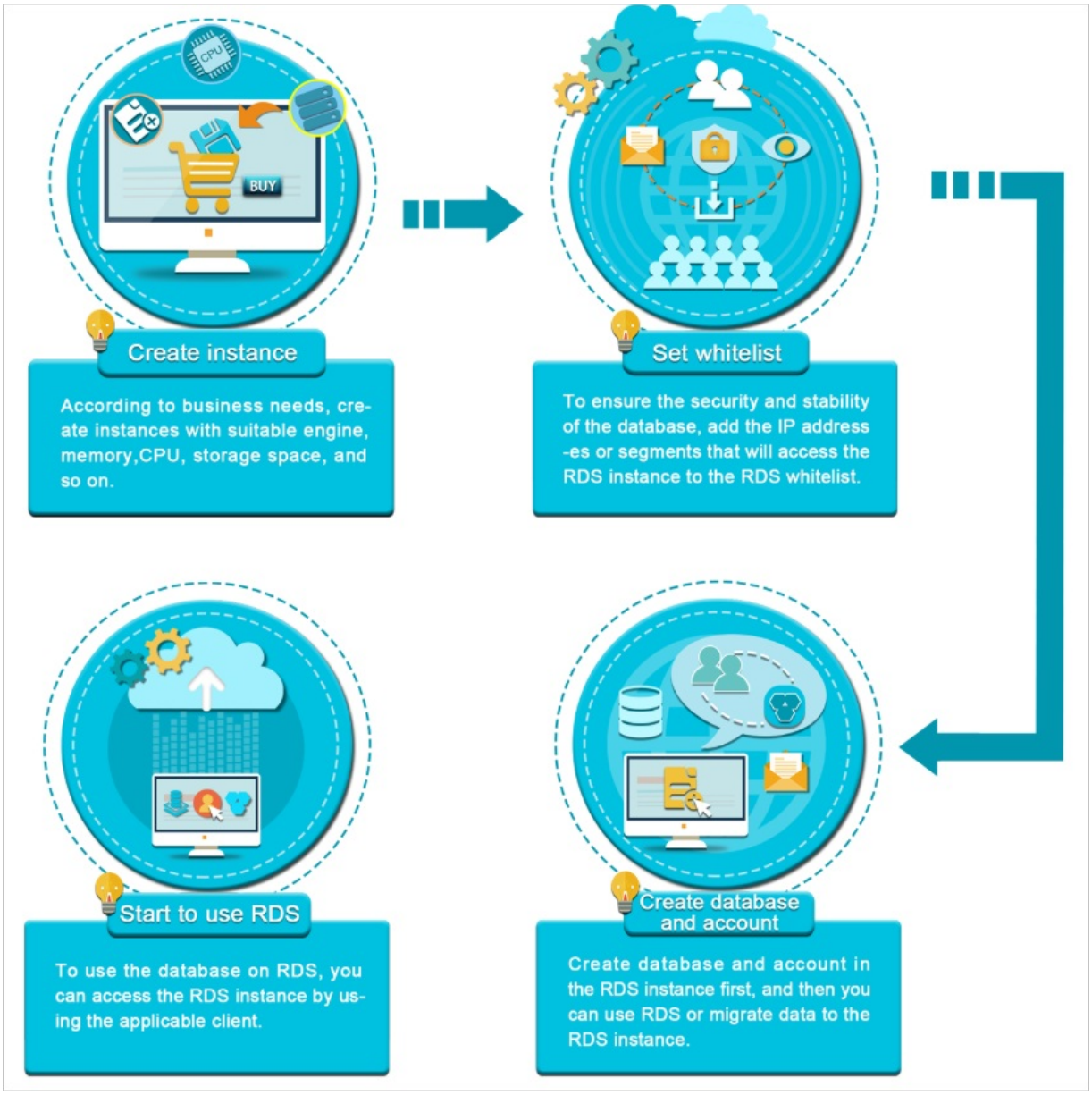
12.1.4.1. Procedure

ApsaraDB RDS quick start covers the following topics: creating an ApsaraDB RDS instance, configuring a whitelist, creating a database, creating an account, and connecting to the instance.

Flowchart for an ApsaraDB RDS instance

If you are using ApsaraDB RDS for the first time, you can start with Limits.

The following figure shows the operations that you must perform before you use an ApsaraDB RDS instance.



12.1.4.2. Create an instance

This topic describes how to create one or more ApsaraDB RDS for PostgreSQL instances in the ApsaraDB RDS console.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic	**Organization**	The organization to which the instance belongs.

Settings Section	Parameter	Description
	Resource Set	The resource set to which the instance belongs.
Region	**Region**	The region in which you want to create the instance. Services in different regions cannot communicate over an internal network. After the instance is created, the region cannot be changed.
	Zone of Primary Node	The zone where the primary instance is deployed.
	Deployment Method	Specifies whether to deploy the primary and secondary instances in separate zones. ApsaraDB RDS supports **Multi-zone Deployment** and **Single-zone Deployment**. If you select **Multi-zone Deployment**, you must configure **Zone of Secondary Node**.
	Zone of Secondary Node	The zone where the secondary instance is deployed. This parameter is available only when **Deployment Method** is set to **Multi-zone Deployment**. **Note** If you select the same zone for both the primary and secondary instances, the deployment is equivalent to single-zone deployment.
Specifications	**Quantity**	The number of ApsaraDB RDS instances that you want to create. Default value: 1.
	Instance Name	The name of the instance. ○ The name must be 2 to 64 characters in length. ○ The name must start with a letter. ○ The name can contain letters, digits, and the following special characters: _ - : ○ The name cannot start with http:// or https://.
	Network Type	The connection type of the instance. ApsaraDB RDS instances support the following connection types: ○ **Internet Connection**: ApsaraDB RDS instances of this connection type can be connected over the Internet. ○ **Internal Network**: ApsaraDB RDS instances of this connection type can be connected over an internal network. **Note** The value of this parameter cannot be changed after the instance is created. Proceed with caution.
	Database Engine	The database engine of the instance. Select **PostgreSQL**.
	Engine Version	The version of the database engine. Valid values: ○ 9.4 ○ 10.0 ○ 11.0 ○ 12.0

Section	Parameter	Description
	Edition	The edition of the instance. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Type	The storage type of the instance. Local and standard SSDs are supported. **Note** PostgreSQL 9.4 instances support only local SSDs.
	Encrypted	Specifies whether to encrypt the standard SSD. This parameter is available only when **Storage Type** is set to **Standard SSD**. If you select Encrypted, you must specify the **Key** parameter. If you do not have a key, you must first create one in Key Management Service (KMS). For more information, see Create a CMK in *KMS User Guide*.
	Key	The key used to encrypt the standard SSD. This parameter is available only when you select **Encrypted**.
	Instance Type	The instance type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Capacity	The storage capacity of the instance, which includes the space to store data, system files, binlog files, and transaction files. Valid values: 20 to 600. Unit: GB. The value must be in 1 GB increments.
Network	**Network Type**	The network type of the instance. ApsaraDB RDS instances support the following network types: ○ **Classic Network**: Cloud services in the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ○ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for enhanced security. **Note** If you select VPC, you must specify a VPC and a vSwitch.
	IP Address Whitelist	The IP addresses that are allowed to connect to the ApsaraDB RDS instance.

4. Click **Submit**.

12.1.4.3. Configure an IP address whitelist

This topic describes how to configure a whitelist for an ApsaraDB RDS instance. Only entities that are listed in a whitelist can access your ApsaraDB RDS instance.

Context

Whitelists make your ApsaraDB RDS instance more secure and do not interrupt the operations of your ApsaraDB RDS instance when you configure whitelists. We recommend that you perform maintenance on your whitelists on a regular basis.

To configure a whitelist, perform the following operations:

- Configure a whitelist: Add IP addresses to allow them to connect to the ApsaraDB RDS instance.

> **Note** The IP address whitelist labeled default contains only the default IP address 0.0.0.0/0, which allows all entities to access your ApsaraDB RDS instance.

- Configure an ECS security group: Add an ECS security group for the ApsaraDB RDS instance to allow ECS instances in the group to connect to the ApsaraDB RDS instance.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Edit** corresponding to the **default** whitelist.

   > **Note** You can also click **Create Whitelist** to create a whitelist.

6. In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks used to access the instance and click **OK**. The following section describes the rules:
   - If you enter the CIDR block 10.10.10.0/24 in the IP Addresses field, all IP addresses in the 10.10.10.X format can access your ApsaraDB RDS instance.
   - If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
   - If you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all ECS instances created within your Alibaba Cloud account are displayed. You can select the required IP addresses to add them to the IP address whitelist.

12.1.4.4. Create a database and an account

Before you start to use ApsaraDB RDS, you must create databases and accounts on an ApsaraDB RDS instance. This topic describes how to create a database and an account on an ApsaraDB RDS for PostgreSQL instance.

Account types

ApsaraDB RDS for PostgreSQL instances support two types of accounts: privileged accounts and standard accounts. The following table describes these account types.

Account type	Description
Privileged account	• You can create and manage privileged accounts only by using the ApsaraDB RDS console or API operations. • If your ApsaraDB RDS instance uses local SSDs, you can create only a single privileged account. If your ApsaraDB RDS instance uses standard or enhanced SSDs, you can create more than one privileged account. A privileged account allows you to manage all the standard accounts and databases that are created on your ApsaraDB RDS instance. • A privileged account has more permissions that allow you to manage your ApsaraDB RDS instance at more fine-grained levels. For example, you can grant the query permissions on different tables to different users. • A privileged account has the permissions to disconnect accounts that are created on your ApsaraDB RDS instance.

Account type	Description
Standard account	• You can create and manage standard accounts by using the ApsaraDB RDS console, API operations, or SQL statements. • You can create more than one standard account on your ApsaraDB RDS instance. • You must grant the permissions on specific databases to a standard account. • A standard account does not have the permissions to create, manage, or disconnect other accounts on your ApsaraDB RDS instance.

Precautions

- If your ApsaraDB RDS instance uses local SSDs, you can create one privileged account in the ApsaraDB RDS console. After the privileged account is created, it cannot be deleted. You can also create and manage more than one standard account by using SQL statements.
- If your ApsaraDB RDS instance uses standard or enhanced SSDs, you can create more than one privileged account and standard account in the ApsaraDB RDS console. You can also create and manage more than one standard account by using SQL statements.
- To migrate data from an on-premises database to your ApsaraDB RDS instance, you must create a database and an account on the ApsaraDB RDS instance. Make sure that the created database has the same properties as the on-premises database. Also make sure that the created account has the same permissions on the created database as the account that is authorized to manage the on-premises database.
- Follow the least privilege principle to create accounts and grant them read-only permissions or read and write permissions on databases. If necessary, you can create more than one account and grant them only the permissions on specific databases. If an account does not need to write data to a database, grant only the read-only permissions on that database to the account.
- For security purposes, we recommend that you specify strong passwords for the accounts on your ApsaraDB RDS instance and change the passwords on a regular basis.

Create a privileged account on an instance that uses local SSDs

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. On the Accounts page, click **Create Privileged Account** and configure the following parameters.

Parameter	Description
Database Account	◦ The name of the account must be 2 to 16 characters in length. ◦ The name can contain lowercase letters, digits, and underscores (_). ◦ The name must start with a letter and end with a letter or digit.
Password	◦ The password of the account must be 8 to 32 characters in length. ◦ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ◦ Special characters include ! @#$%^&*()_+-=
Re-enter Password	Enter the password of the account again.

6. Click **Create**.

Create a privileged or standard account on an instance that uses standard or enhanced SSDs

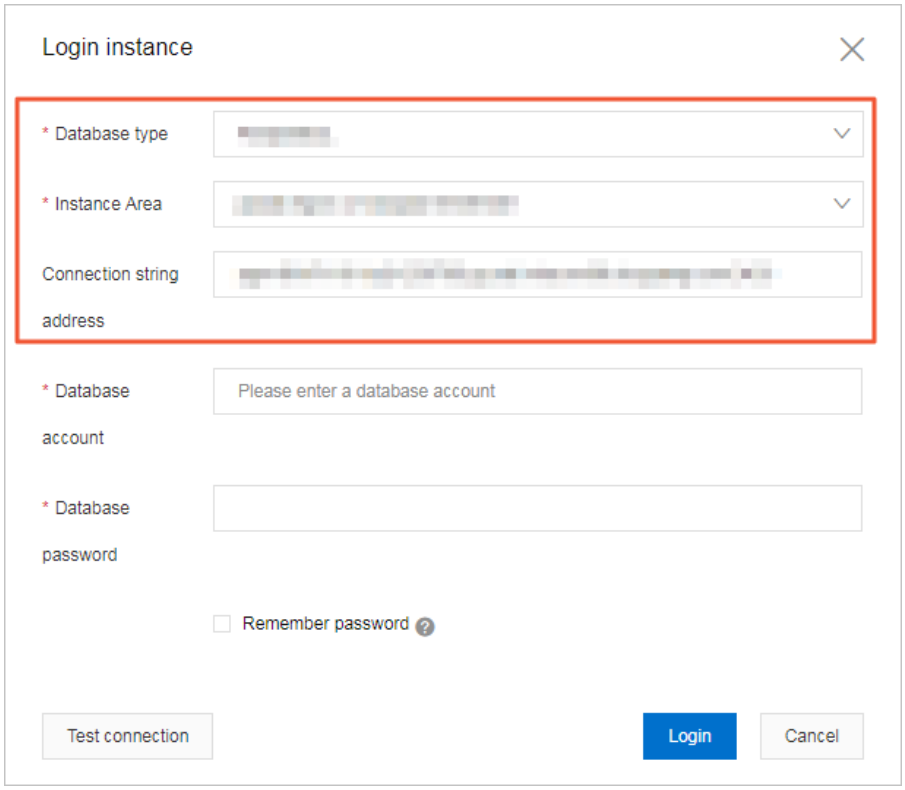
1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. On the Accounts page, click **Create Account** and configure the following parameters.

Parameter	Description
Database Account	○ The name of the account must be 2 to 16 characters in length. ○ The name can contain lowercase letters, digits, and underscores (_). ○ The name must start with a letter and end with a letter or digit.
Account Type	Select **Privileged Account** or **Standard Account**.
Password	○ The password of the account must be 8 to 32 characters in length. ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ Special characters include ! @#$%^&*()_+-=
Re-enter Password	Enter the password of the account again.
Description	This parameter is optional. You can enter relevant description to make the instance identifiable. The description can be up to 256 characters in length.

6. Click **Create**.

Create a database and a standard account

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database password**, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**. If you want the browser to remember the password, select **Remember password** before you click **Login**.

> **Note** If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see Configure an IP address whitelist.

7. The **SQLConsole** page appears after you log on to the instance. Execute a statement in the following format to create a database:

```
CREATE DATABASE name
 [ [ WITH ] [ OWNER [=] user_name ]
        [ TEMPLATE [=] template ]
        [ ENCODING [=] encoding ]
        [ LC_COLLATE [=] lc_collate ]
        [ LC_CTYPE [=] lc_ctype ]
        [ TABLESPACE [=] tablespace_name ]
        [ CONNECTION LIMIT [=] connlimit ] ]
```

For example, if you want to create a database named test, execute the following statement:

```
create database test;
```

8. Click **execute**.
9. In the SQL window, execute a statement in the following format to create a standard account:

```
CREATE USER name [ [ WITH ] option [ ... ] ]
where option can be:
   SUPERUSER | NOSUPERUSER
 | CREATEDB | NOCREATEDB
 | CREATEROLE | NOCREATEROLE
 | CREATEUSER | NOCREATEUSER
 | INHERIT | NOINHERIT
 | LOGIN | NOLOGIN
 | REPLICATION | NOREPLICATION
 | CONNECTION LIMIT connlimit
 | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
 | VALID UNTIL 'timestamp'
 | IN ROLE role_name [, ...]
 | IN GROUP role_name [, ...]
 | ROLE role_name [, ...]
 | ADMIN role_name [, ...]
 | USER role_name [, ...]
 | SYSID uid
```

For example, if you want to create a user account named test2 whose password is 123456, execute the following statement:

```
create user test2 password '123456';
```

10. Click **execute**.

12.1.4.5. Connect to an ApsaraDB RDS for PostgreSQL instance

This topic describes how to use Data Management (DMS) or the pgAdmin 4 client to connect to an ApsaraDB RDS instance.

Context

You can log on to DMS from the ApsaraDB RDS console and then connect to an ApsaraDB RDS instance.

DMS is a data management service that integrates data, schema, and server management, access security, BI charts, data trends, data tracking, and performance optimization. DMS can be used to manage relational and non-relational databases, such as MySQL, SQL Server, PostgreSQL, MongoDB, and Redis. It can also be used to manage Linux servers.

You can also use a client to connect to an ApsaraDB RDS instance. ApsaraDB RDS for PostgreSQL is fully compatible with PostgreSQL. You can connect to an ApsaraDB RDS for PostgreSQL instance in a similar manner as you would connect to an open source PostgreSQL instance. In this topic, the pgAdmin 4 client is used to connect to an ApsaraDB RDS instance.

Use DMS to connect to an ApsaraDB RDS instance

For more information about how to use DMS to connect to an ApsaraDB RDS instance, see Log on to an ApsaraDB for RDS instance by using DMS.

Use the pgAdmin 4 client to connect to an ApsaraDB RDS instance

1. Add the IP address of the pgAdmin client to an IP address whitelist of the ApsaraDB RDS instance. For more information about how to configure a whitelist, see Configure an IP address whitelist.
2. Start the pgAdmin 4 client.

   > **Note** For information about how to download the pgAdmin 4 client, visit pgAdmin 4 (Windows).

3. Right-click **Servers** and choose **Create > Server**, as shown in the following figure.
4. On the **General** tab of the **Create - Server** dialog box, enter the name of the server, as shown in the following figure.
5. Click the **Connection** tab and enter the information of the instance, as shown in the following figure.

Parameter	Description
Host name/address	The internal endpoint of the ApsaraDB RDS instance. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.
Port	The internal port number that is used to connect to the ApsaraDB RDS instance. For more information about how to view the internal port number, see View and modify the internal endpoint and port number.
Username	The name of the privileged account on the ApsaraDB RDS instance. For more information about how to obtain a privileged account, see Create a database and an account.
Password	The password of the privileged account of the ApsaraDB RDS instance.

6. Click **Save**.
7. If the connection information is correct, choose **Servers > Server Name > Databases > postgres**. If the following page appears, the connection is established.

   > **Notice** The postgres database is the default system database of the ApsaraDB RDS instance. Do not perform operations on this database.

12.1.5. Instances

12.1.5.1. Create an instance

This topic describes how to create one or more ApsaraDB RDS for PostgreSQL instances in the ApsaraDB RDS console.

Procedure

1. Log on to the ApsaraDB RDS console.

2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	**Organization**	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	**Region**	The region in which you want to create the instance. Services in different regions cannot communicate over an internal network. After the instance is created, the region cannot be changed.
	Zone of Primary Node	The zone where the primary instance is deployed.
	Deployment Method	Specifies whether to deploy the primary and secondary instances in separate zones. ApsaraDB RDS supports **Multi-zone Deployment** and **Single-zone Deployment**. If you select **Multi-zone Deployment**, you must configure **Zone of Secondary Node**.
	Zone of Secondary Node	The zone where the secondary instance is deployed. This parameter is available only when **Deployment Method** is set to **Multi-zone Deployment**. **Note** If you select the same zone for both the primary and secondary instances, the deployment is equivalent to single-zone deployment.
	Quantity	The number of ApsaraDB RDS instances that you want to create. Default value: 1.
	Instance Name	The name of the instance. ○ The name must be 2 to 64 characters in length. ○ The name must start with a letter. ○ The name can contain letters, digits, and the following special characters: _ - : ○ The name cannot start with http:// or https://.
	Network Type	The connection type of the instance. ApsaraDB RDS instances support the following connection types: ○ **Internet Connection**: ApsaraDB RDS instances of this connection type can be connected over the Internet. ○ **Internal Network**: ApsaraDB RDS instances of this connection type can be connected over an internal network. **Note** The value of this parameter cannot be changed after the instance is created. Proceed with caution.

Section	Parameter	Description
Specifications	Database Engine	The database engine of the instance. Select **PostgreSQL**.
	Engine Version	The version of the database engine. Valid values: ○ 9.4 ○ 10.0 ○ 11.0 ○ 12.0
	Edition	The edition of the instance. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Type	The storage type of the instance. Local and standard SSDs are supported. **Note** PostgreSQL 9.4 instances support only local SSDs.
	Encrypted	Specifies whether to encrypt the standard SSD. This parameter is available only when **Storage Type** is set to **Standard SSD**. If you select Encrypted, you must specify the **Key** parameter. If you do not have a key, you must first create one in Key Management Service (KMS). For more information, see Create a CMK in *KMS User Guide*.
	Key	The key used to encrypt the standard SSD. This parameter is available only when you select **Encrypted**.
	Instance Type	The instance type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Capacity	The storage capacity of the instance, which includes the space to store data, system files, binlog files, and transaction files. Valid values: 20 to 600. Unit: GB. The value must be in 1 GB increments.
Network	Network Type	The network type of the instance. ApsaraDB RDS instances support the following network types: ○ **Classic Network**: Cloud services in the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ○ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for enhanced security. **Note** If you select VPC, you must specify a VPC and a vSwitch.
	IP Address Whitelist	The IP addresses that are allowed to connect to the ApsaraDB RDS instance.

4. Click **Submit**.

12.1.5.2. Create an ApsaraDB RDS for PostgreSQL instance that uses standard or enhanced SSDs

Cloud disks are block-level data storage products provided by Alibaba Cloud for ECS. They provide low latency, high performance, durability, and reliability. This topic describes how to create one or more instances that use standard or enhanced SSDs in the ApsaraDB RDS console.

Prerequisites

The instance runs PostgreSQL 10.0 or later.

Context

An ApsaraDB RDS instance with standard or enhanced SSDs uses a distributed triplicate mechanism to ensure 99.9999999% data reliability. If service disruptions occur within a zone due to hardware failure, data in that zone is copied to an unaffected disk in another zone to ensure data availability.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	**Organization**	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	**Region**	The region in which you want to create the instance. Services in different regions cannot communicate over an internal network. After the instance is created, the region cannot be changed.
	Zone of Primary Node	The zone where the primary instance is deployed.
	Deployment Method	Specifies whether to deploy the primary and secondary instances in separate zones. ApsaraDB RDS supports **Multi-zone Deployment** and **Single-zone Deployment**. If you select **Multi-zone Deployment**, you must configure **Zone of Secondary Node**.
	Zone of Secondary Node	The zone where the secondary instance is deployed. This parameter is available only when **Deployment Method** is set to **Multi-zone Deployment**. **Note** If you select the same zone for both the primary and secondary instances, the deployment is equivalent to single-zone deployment.
	Quantity	The number of ApsaraDB RDS instances that you want to create. Default value: 1.

Section	Parameter	Description
Specifications	Instance Name	The name of the instance. ○ The name must be 2 to 64 characters in length. ○ The name must start with a letter. ○ The name can contain letters, digits, and the following special characters: _ - : ○ The name cannot start with http:// or https://.
	Network Type	The connection type of the instance. ApsaraDB RDS instances support the following connection types: ○ **Internet Connection**: ApsaraDB RDS instances of this connection type can be connected over the Internet. ○ **Internal Network**: ApsaraDB RDS instances of this connection type can be connected over an internal network. **Note** The value of this parameter cannot be changed after the instance is created. Proceed with caution.
	Database Engine	The database engine of the instance. Select **PostgreSQL**.
	Engine Version	The version of the database engine. Set the value to **10.0** or a later version.
	Edition	The edition of the instance. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Type	The storage type of the instance. Set the value to **Standard SSD**.
	Encrypted	Specifies whether to encrypt the standard SSD. If you select Encrypted, you must specify the **Key** parameter. If you do not have a key, you must first create one in Key Management Service (KMS). For more information, see Configure data encryption. **Note** Disk encryption provides maximum protection for your data with minimal impact on your businesses or applications. Both the snapshots generated from encrypted disks and the disks created from those snapshots are automatically encrypted.
	Key	The key used to encrypt the standard SSD. This parameter is available only when you select **Encrypted**.
	Instance Type	The instance type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. For more information, see Instance types in *ApsaraDB RDS Product Introduction*.
	Storage Capacity	The storage capacity of the instance, which includes the space to store data, system files, binlog files, and transaction files. Valid values: 20 to 600. Unit: GB. The value must be in 1 GB increments.

Section	Parameter	Description
Network	Network Type	The network type of the instance. ApsaraDB RDS instances support the following network types: ○ **Classic Network**: Cloud services in the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ○ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for enhanced security. **Note** If you select VPC, you must specify a VPC and a vSwitch.
	IP Address Whitelist	The IP addresses that are allowed to connect to the ApsaraDB RDS instance.

4. Click **Submit**.

12.1.5.3. View basic information of an instance

This topic describes how to view the details of an ApsaraDB RDS instance, such as basic information, internal network connection information, status, and configurations.

Procedure

1. Log on to the ApsaraDB RDS console.
2. Use one of the following methods to go to the **Basic Information** page of an instance:
   - On the **Instances** page, click the ID of the instance to go to the **Basic Information** page.
   - On the **Instances** page, find the instance and click **Manage** in the corresponding **Actions** column. The **Basic Information** page appears.

12.1.5.4. Restart an instance

This topic describes how to manually restart an ApsaraDB RDS instance. This applies if the number of connections exceeds the specified threshold or if an instance has performance issues.

Prerequisites

The instance is in the **Running** state.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Restart Instance**.

Note When you restart an instance, applications are disconnected from the instance. We recommend that you make appropriate service arrangements before you restart an instance. Proceed with caution.

5. In the message that appears, click **Confirm**.

12.1.5.5. Change the specifications of an instance

This topic describes how to change the specifications of an ApsaraDB RDS instance. You can upgrade or downgrade an ApsaraDB RDS instance to meet your business needs.

Prerequisites

The instance is in the **Running** state and is not in the backing up or restoring state.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configure Information** section of the **Basic Information** page, click **Change Specifications**.
5. On the **Change Specifications** page, set **Edition**, **Instance Type**, and **Storage Capacity**.
6. Click **Submit**.

12.1.5.6. Set a maintenance window

This topic describes how to set a maintenance window for an ApsaraDB RDS instance.

Context

The backend system performs maintenance on the ApsaraDB RDS instances during the maintenance window. This ensures the stability of the ApsaraDB RDS instance. The default maintenance window is from 02:00 (UTC+8) to 06:00 (UTC+8). We recommend that you set the maintenance window to off-peak hours of your business to avoid impacts on your business.

Precautions

- An instance enters the **Maintaining Instance** state before the maintenance window to ensure stability during the maintenance process. When the instance is in this state, access to data in the database and query operations such as performance monitoring are not affected. However, except for account and database management and IP address whitelist configuration, modification operations such as upgrade, downgrade, and restart are temporarily unavailable.
- During the maintenance window, one or two network interruptions may occur. Make sure that your applications are configured with automatic reconnection policies.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section, click **Configure** to the right of **Maintenance Window**.
5. Select a maintenance window and click **Save**.

> **Note** The maintenance window is displayed in UTC+8.

12.1.5.7. Configure primary/secondary switchover

ApsaraDB RDS provides the primary/secondary switchover feature to ensure the high availability of databases. The primary/secondary switchover is performed when the primary instance becomes unavailable. You can also manually switch your business to the secondary instance. This topic describes how to manually switch over services between a primary instance and its secondary instance.

Context

Data is synchronized in real time between the primary and secondary instances. You can access only the primary instance. The secondary instance serves only as a backup instance and does not allow external access. After the switchover, the original primary instance becomes the secondary instance.

> **Note** Network interruptions may occur during a switchover. Make sure that your applications are configured with automatic reconnection policies.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Service Availability**.
5. In the **Availability Information** section, click **Switch Primary/Secondary Instance**.
6. In the **Switch Primary/Secondary Instance** message, click **OK**.

> **Note**
> - During the switchover, operations such as managing databases and accounts and changing network types cannot be performed. Therefore, we recommend that you select Switch Within Maintenance Window.
> - For more information about how to set a maintenance window, see Set a maintenance window.

12.1.5.8. Release an instance

This topic describes how to manually release an instance.

Precautions

- Only instances in the running state can be released.
- After an instance is released, the instance data is immediately deleted. We recommend that you back up your data before you release an instance.
- When you release a primary instance, all of its read-only instances are also released.

Procedure

1. Log on to the ApsaraDB RDS console.
2. Find the instance that you want to release and choose **More > Release Instance** in the Actions column.
3. In the **Release Instance** message, click **Confirm**.

12.1.5.9. Modify parameters of an instance

This topic describes how to view and modify the values of some parameters and query parameter modification records in the console.

Precautions

- To ensure instance stability, you can modify only specific parameters in the ApsaraDB RDS console.
- When you modify parameters on the **Editable Parameters** tab, refer to the **Value Range** column corresponding to each parameter.
- After specific parameters are modified, you must restart your instance for the changes to take effect. The necessity of restart is displayed in the **Force Restart** column on the **Editable Parameters** tab. We recommend that you modify the parameters of an instance during off-peak hours and make sure that your applications are configured with automatic reconnection policies.

Modify parameters

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Parameters**.
5. Perform the following operations:

   Export the parameter settings of the instance to your computer.

   On the Editable Parameters tab, click **Export Parameters**. The parameter settings of the instance are exported as a TXT file to your computer.

   Modify and import the parameter settings.

   i. After you modify parameters in the exported parameter file, click **Import Parameters** and copy the parameter settings to the field.

   ii. Click **OK**.

   iii. In the upper-right corner of the page, click **Apply Changes**.

   > **Note**
   > - If the new parameter value takes effect only after an instance restart, the system prompts you to restart the instance. We recommend that you restart the instance during off-peak hours and make sure that your applications are configured with automatic reconnection policies.
   > - Before the new parameter values are applied, you can click **Cancel Changes** to cancel them.

   Modify a single parameter.

   i. On the **Editable Parameters** tab, find the parameter that you want to modify and click the icon in the **Actual Value** column.

   ii. Enter a new value based on the prompted value range.

   iii. Click **Confirm**.

iv. In the upper-right corner of the page, click **Apply Changes**.

> **Note**
> - If the new parameter value takes effect only after an instance restart, the system prompts you to restart the instance. We recommend that you restart the instance during off-peak hours and make sure that your applications are configured with automatic reconnection policies.
> - Before the new parameter value is applied, you can click **Cancel Changes** to cancel it.

View the parameter modification history

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Parameters**.
5. On the page that appears, click the **Edit History** tab.
6. Select a time range and click **Search**.

12.1.5.10. Read-only instances

12.1.5.10.1. Overview of read-only ApsaraDB RDS for PostgreSQL instances

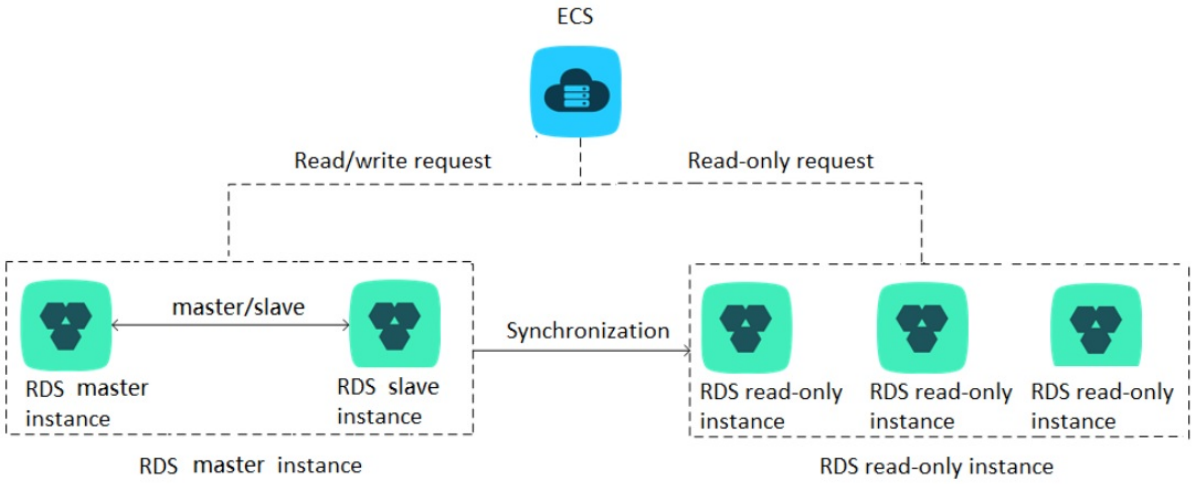
This topic provides an overview of read-only ApsaraDB RDS for PostgreSQL instances. If a large number of read requests overwhelm the primary instance, your business may be interrupted. In this case, you can create one or more read-only instances to offload read requests from the primary instance. This scales the read capability of your database system and increases the throughput of your application.

Overview

When a read-only instance is created, the data is replicated from the secondary instance. The data is consistent with that of the primary instance. Data updates of the primary instance are automatically synchronized to all read-only instances immediately after the primary instance completes operations.

> **Note**
> - Only ApsaraDB RDS instances that run PostgreSQL 10.0 support read-only instances.
> - The specifications of the primary instance must have at least eight CPU cores and 32 GB of memory.
> - Each read-only instance works in a single-node architecture, where no instances are provided as backups.

The following figure shows the topology of read-only instances.



Features

- Region and zone: Read-only instances reside within the same region as the primary instance, but possibly in different zones.
- Specifications and storage space: The specifications and storage space of read-only instances cannot be lower than those of the primary instance.
- Network type: The network type of a read-only instance can differ from that of the primary instance.
- Account and database management: Read-only instances do not require database or account maintenance, because their database and account information is synchronized with the primary instance.
- IP address whitelist: A read-only instance automatically replicates the IP address whitelists of the primary instance. However, the IP address whitelists for the read-only instance are independent of those of the primary instance. For information about how to modify the IP address whitelists of a read-only instance, see Configure an IP address whitelist.
- Monitoring and alerts: You can monitor system performance metrics, such as the disk capacity, IOPS, number of connections, and CPU utilization.

Limits

- Number of read-only instances: A maximum of five read-only instances can be created on a primary instance.
- Instance backup: Read-only instances do not support backup settings or manual backups because backups have been configured or created on the primary instance.
- Data migration: You cannot migrate data to read-only instances.
- Database management: You cannot create or delete databases on read-only instances.
- Account management: You cannot create or delete accounts, authorize accounts, or change the passwords of accounts on read-only instances.

FAQ

Q: Can I manage accounts created on the primary instance from its read-only instances?

A: No, although accounts created on the primary instance are replicated to its read-only instances, you cannot manage the accounts on the read-only instances. The accounts have only read permissions on the read-only instances.

12.1.5.10.2. Create a read-only ApsaraDB RDS for PostgreSQL instance

This topic describes how to create a read-only instance for your primary ApsaraDB RDS for PostgreSQL instance. This allows your database system to process a large number of read requests and increases the throughput of your application. The data on each read-only instance is a copy of that of the primary instance. Data updates to the primary instance are synchronized to all of its read-only instances.

Prerequisites

- The primary instance runs PostgreSQL 10.0.
- The specifications of the primary instance must have at least eight CPU cores and 32 GB of memory.

Precautions

- You can create read-only instances only for your primary instance. You cannot change existing instances to read-only instances.
- When you create a read-only instance, the system replicates data from a secondary instance. Therefore, operations on your primary instance are not interrupted.
- A read-only instance does not inherit the parameter settings of the primary instance. The system generates default parameter settings for the read-only instance, and you can modify the settings in the ApsaraDB RDS console.
- The instance type and storage capacity of a read-only instance cannot be lower than that of the primary instance.
- You can create up to five read-only instances.

Create a read-only instance

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the Distributed by Instance Role section of the Basic Information page, click **Create Read-only Instance**.
5. On the Create Read-only Instance page, configure parameters and click **Submit**. The following table describes the parameters.

Section	Parameter	Description
Region	Region	The region where the instance is deployed.
Specifications	Database Engine	The database engine of the read-only instance, which is the same as that of the primary instance and cannot be changed.
	Engine Version	The engine version of the read-only instance, which is the same as that of the primary instance and cannot be changed.
	Edition	The edition of the read-only instance, which is the same as that of the primary instance and cannot be changed.
	Instance Type	The instance type of the read-only instance. The instance type of the read-only instance can be different from that of the primary instance, and can be changed at any time to facilitate flexible upgrade and downgrade. To ensure sufficient I/O throughput for data synchronization, we recommend that you select at least the same instance type as the primary instance for the read-only instance.

Section	Parameter	Description
	Storage Capacity	The storage capacity of the read-only instance. To ensure sufficient I/O throughput for data synchronization, we recommend that you select at least the same storage capacity as the primary instance for the read-only instance.
Network Type	Network Type	The network type of the read-only instance, which is the same as that of the primary instance and cannot be changed.
	VPC	Select a VPC if the network type is set to VPC.
	vSwitch	Select a vSwitch if the network type is set to VPC.

12.1.5.10.3. View a read-only ApsaraDB RDS for PostgreSQL instance

This topic describes how to view details of a read-only ApsaraDB RDS for PostgreSQL instance. You can go to the Basic Information page of a read-only instance from the Instances page or the read-only instance list of the primary instance. Read-only instances are managed in the same manner as primary instances. The Basic Information page shows the management operations that can be performed.

View instance details of a read-only instance by using its ID

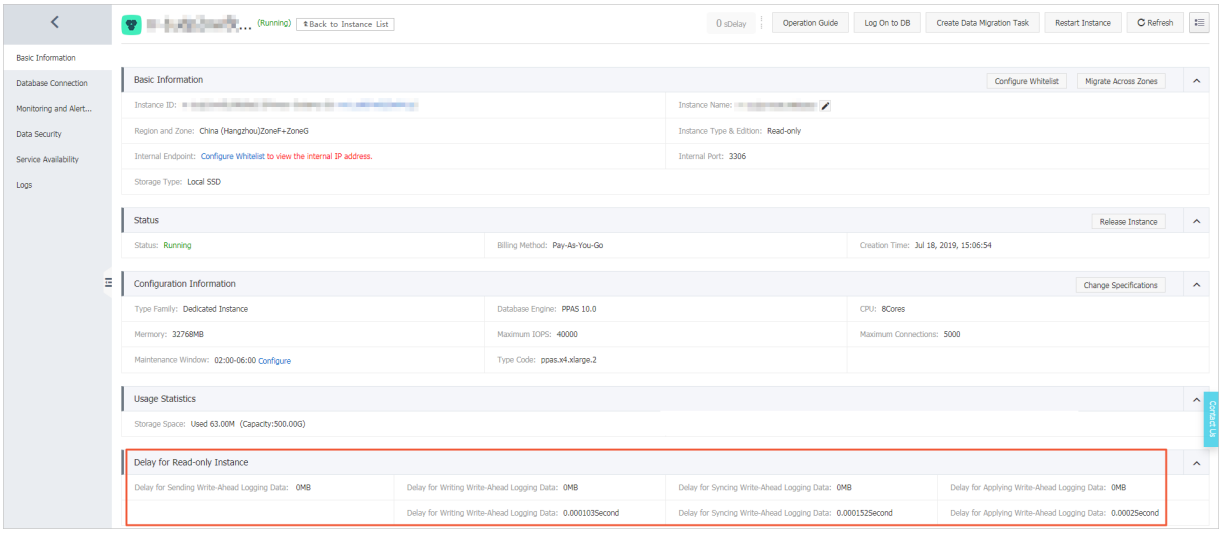
1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the instance that you want to view.
3. Click the ID of the instance or click **Manage** in the corresponding Actions column to go to the Basic Information page.

View details of a read-only instance by using the primary instance

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. On the **Basic Information** page, move the pointer over the number below **Read-only Instance** in the **Distributed by Instance Role** section. The ID of the read-only instance is displayed.
5. Click the ID of the read-only instance to go to the Basic Information page of the read-only instance.

View the latency of a read-only instance

When a read-only instance synchronizes data from its primary RDS instance, latency may occur. You can navigate to the Basic Information page of the read-only instance to view the latency of data synchronization to the instance.



12.1.6. Database connection

12.1.6.1. Connect to an ApsaraDB RDS for PostgreSQL instance

This topic describes how to use Data Management (DMS) or the pgAdmin 4 client to connect to an ApsaraDB RDS instance.

Context

You can log on to DMS from the ApsaraDB RDS console and then connect to an ApsaraDB RDS instance.

DMS is a data management service that integrates data, schema, and server management, access security, BI charts, data trends, data tracking, and performance optimization. DMS can be used to manage relational and non-relational databases, such as MySQL, SQL Server, PostgreSQL, MongoDB, and Redis. It can also be used to manage Linux servers.

You can also use a client to connect to an ApsaraDB RDS instance. ApsaraDB RDS for PostgreSQL is fully compatible with PostgreSQL. You can connect to an ApsaraDB RDS for PostgreSQL instance in a similar manner as you would connect to an open source PostgreSQL instance. In this topic, the pgAdmin 4 client is used to connect to an ApsaraDB RDS instance.

Use DMS to connect to an ApsaraDB RDS instance

For more information about how to use DMS to connect to an ApsaraDB RDS instance, see Log on to an ApsaraDB for RDS instance by using DMS.

Use the pgAdmin 4 client to connect to an ApsaraDB RDS instance

1. Add the IP address of the pgAdmin client to an IP address whitelist of the ApsaraDB RDS instance. For more information about how to configure a whitelist, see Configure an IP address whitelist.
2. Start the pgAdmin 4 client.

   > **Note** For information about how to download the pgAdmin 4 client, visit pgAdmin 4 (Windows).

3. Right-click **Servers** and choose **Create > Server**, as shown in the following figure.
4. On the **General** tab of the **Create - Server** dialog box, enter the name of the server, as shown in the following figure.
5. Click the **Connection** tab and enter the information of the instance, as shown in the following figure.

Parameter	Description
Host name/address	The internal endpoint of the ApsaraDB RDS instance. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.
Port	The internal port number that is used to connect to the ApsaraDB RDS instance. For more information about how to view the internal port number, see View and modify the internal endpoint and port number.
Username	The name of the privileged account on the ApsaraDB RDS instance. For more information about how to obtain a privileged account, see Create a database and an account.
Password	The password of the privileged account of the ApsaraDB RDS instance.

6. Click **Save**.
7. If the connection information is correct, choose **Servers > Server Name > Databases > postgres**. If the following page appears, the connection is established.

> **Notice** The postgres database is the default system database of the ApsaraDB RDS instance. Do not perform operations on this database.

12.1.6.2. Use DMS to log on to an ApsaraDB RDS instance

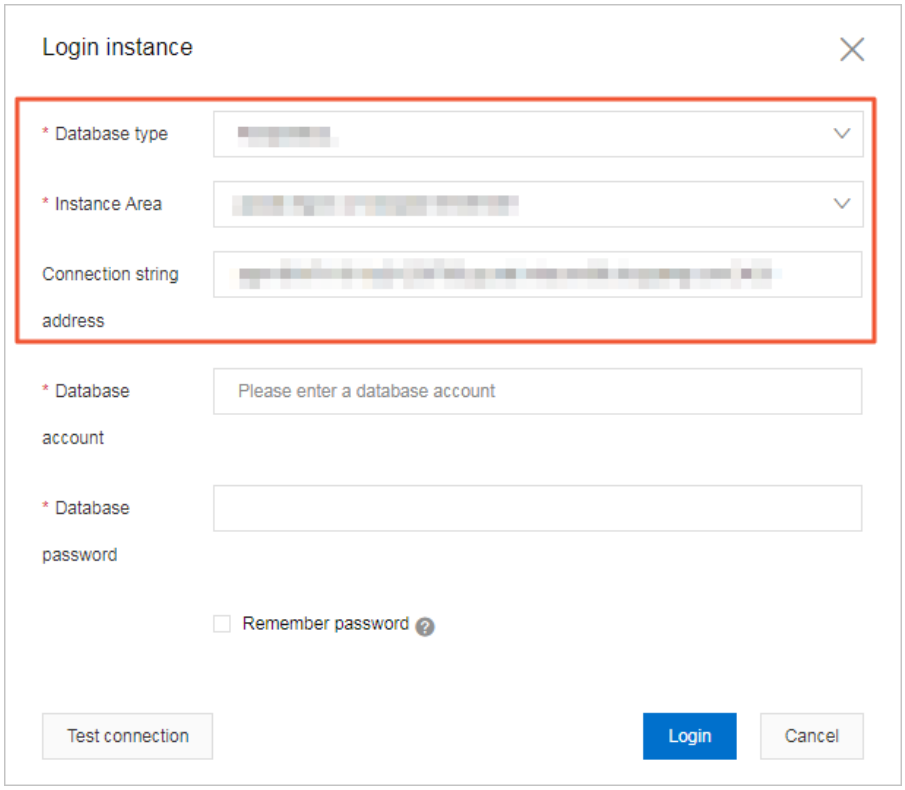
This topic describes how to use Data Management (DMS) to log on to an ApsaraDB RDS instance.

Prerequisites

The IP address whitelist is configured. For more information about how to configure an IP address whitelist, see Configure an IP address whitelist.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database password**, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**.

> **Note** If you want the browser to remember the password, select **Remember password** before you click **Login**.

12.1.6.3. View and modify the internal endpoint and port number

You must use the internal endpoint and port number to access an ApsaraDB RDS instance. This topic describes how to view and modify the internal endpoint and port number of an ApsaraDB RDS instance in the ApsaraDB RDS console.

View the internal endpoint and port number

1. Log on to the ApsaraDB RDS console.

2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the **Basic Information** section, view the internal endpoint and port number of the instance.

Modify the internal endpoint and port number

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. On the right side of the page, click **Change Endpoint**.
6. In the dialog box that appears, set **Connection Type** to **Internal Endpoint**.
7. Modify the endpoint prefix and port number and click **OK**.

FAQ

- Q: Do I need to modify the endpoint or port number in my application after I modify the endpoint or port number of an instance?

  A: Yes, you must modify the endpoint or port number in the application after you modify them. Otherwise, the application cannot connect to databases of the instance.

- Q: Does the modification of the endpoint take effect immediately? Do I need to restart the instance?

  A: No, you do not need to restart the instance. The modification takes effect immediately.

12.1.7. Accounts

12.1.7.1. Create an account

Before you start to use ApsaraDB RDS, you must create an account on an ApsaraDB RDS instance. This topic describes how to create an account on an ApsaraDB RDS for PostgreSQL instance.

Account types

ApsaraDB RDS for PostgreSQL instances support two types of accounts: privileged accounts and standard accounts. The following table describes these account types.

Account type	Description
Privileged account	• You can create and manage privileged accounts only by using the ApsaraDB RDS console or the API. • If your ApsaraDB RDS instance uses local SSDs, you can create only a single privileged account. If your ApsaraDB RDS instance uses standard or enhanced SSDs, you can create more than one privileged account. A privileged account allows you to manage all the standard accounts and databases that are created on your ApsaraDB RDS instance. • A privileged account has more permissions that allow you to manage your ApsaraDB RDS instance at more fine-grained levels. For example, you can grant the query permissions on different tables to different users. • A privileged account has the permissions to disconnect accounts that are created on your ApsaraDB RDS instance.

Account type	Description
Standard account	• You can create and manage standard accounts by using the ApsaraDB RDS console, API operations, or SQL statements. • You can create more than one standard account on your ApsaraDB RDS instance. • You must grant the permissions on specific databases to a standard account. • A standard account does not have the permissions to create, manage, or disconnect other accounts on your ApsaraDB RDS instance.

Precautions

- If your ApsaraDB RDS instance uses local SSDs, you can create one privileged account in the ApsaraDB RDS console. After the privileged account is created, it cannot be deleted. You can also create and manage more than one standard account by using SQL statements.
- If your ApsaraDB RDS instance uses standard or enhanced SSDs, you can create more than one privileged account and standard account in the ApsaraDB RDS console. You can also create and manage more than one standard account by using SQL statements.
- To migrate data from an on-premises database to your ApsaraDB RDS instance, you must create a database and an account on the ApsaraDB RDS instance. Make sure that the created database has the same properties as the on-premises database. Also make sure that the created account has the same permissions on the created database as the account that is authorized to manage the on-premises database.
- Follow the least privilege principle to create accounts and grant them read-only permissions or read and write permissions on databases. If necessary, you can create more than one account and grant them only the permissions on specific databases. If an account does not need to write data to a database, grant only the read-only permissions on that database to the account.
- For security purposes, we recommend that you specify strong passwords for the accounts on your ApsaraDB RDS instance and change the passwords on a regular basis.

Create a privileged account on an instance that uses local SSDs

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. On the Accounts page, click **Create Privileged Account** and configure the following parameters.

Parameter	Description
Database Account	◦ The name of the account must be 2 to 16 characters in length. ◦ The name can contain lowercase letters, digits, and underscores (_). ◦ The name must start with a letter and end with a letter or digit.
Password	◦ The password of the account must be 8 to 32 characters in length. ◦ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.characters. ◦ Special characters include ! @#$%^&*()_+-=

Parameter	Description
Re-enter Password	Enter the password of the account again.

6. Click **Create**.

Create a privileged or standard account on an instance that uses standard or enhanced SSDs

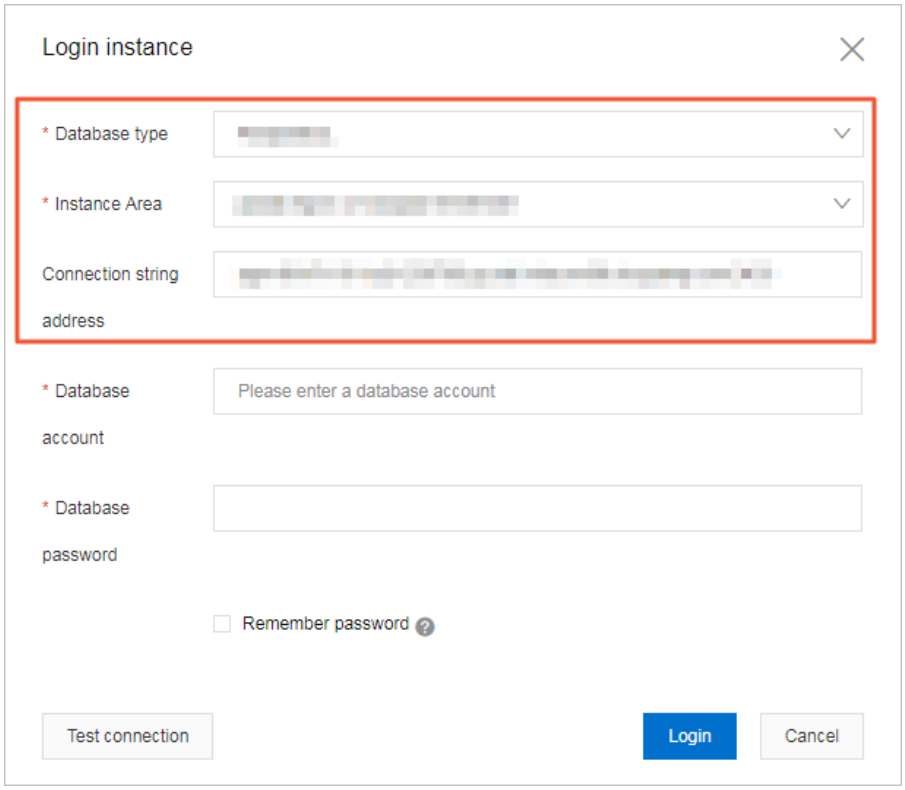
1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. On the Accounts page, click **Create Account** and configure the following parameters.

Parameter	Description
Database Account	○ The name of the account must be 2 to 16 characters in length. ○ The name can contain lowercase letters, digits, and underscores (_). ○ The name must start with a letter and end with a letter or digit.
Account Type	Select **Privileged Account** or **Standard Account**.
Password	○ The password of the account must be 8 to 32 characters in length. ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.characters. ○ Special characters include ! @#$%^&*()_+-=
Re-enter Password	Enter the password of the account again.
Description	This parameter is optional. You can enter relevant description to make the instance identifiable. The description can be up to 256 characters in length.

6. Click **Create**.

Create a standard account on an instance that uses local SSDs

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database password**, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**. If you want the browser to remember the password, select **Remember password** before you click **Login**.

> **Note** If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see Configure an IP address whitelist.

7. The **SQLConsole** page appears after you log on to the instance. Execute a statement in the following format to create a standard account:

```
CREATE USER name [ [ WITH ] option [ ... ] ]
where option can be:
    SUPERUSER | NOSUPERUSER
  | CREATEDB | NOCREATEDB
  | CREATEROLE | NOCREATEROLE
  | CREATEUSER | NOCREATEUSER
  | INHERIT | NOINHERIT
  | LOGIN | NOLOGIN
  | REPLICATION | NOREPLICATION
  | CONNECTION LIMIT connlimit
  | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
  | VALID UNTIL 'timestamp'
  | IN ROLE role_name [, ...]
  | IN GROUP role_name [, ...]
  | ROLE role_name [, ...]
  | ADMIN role_name [, ...]
  | USER role_name [, ...]
  | SYSID uid
```

For example, if you want to create a user account named test2 whose password is 123456, execute the following statement:

```
create user test2 password '123456';
```

8. Click **execute**.

12.1.7.2. Reset the password

This topic describes how to use the ApsaraDB RDS console to reset the password of your database account if you forget the password.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. In the **Actions** column corresponding to the account, click **Reset Password**.
6. In the dialog box that appears, enter a new password and click **OK**.

> **Note** The password must meet the following requirements:
>
> - The password must be 8 to 32 characters in length.
> - The password must contain at least three of the following characters: uppercase letters, lowercase letters, digits, and special characters.
> - Special characters include ! @ # $ % ^ & * ( ) _ + - =

12.1.8. Databases

12.1.8.1. Create a database

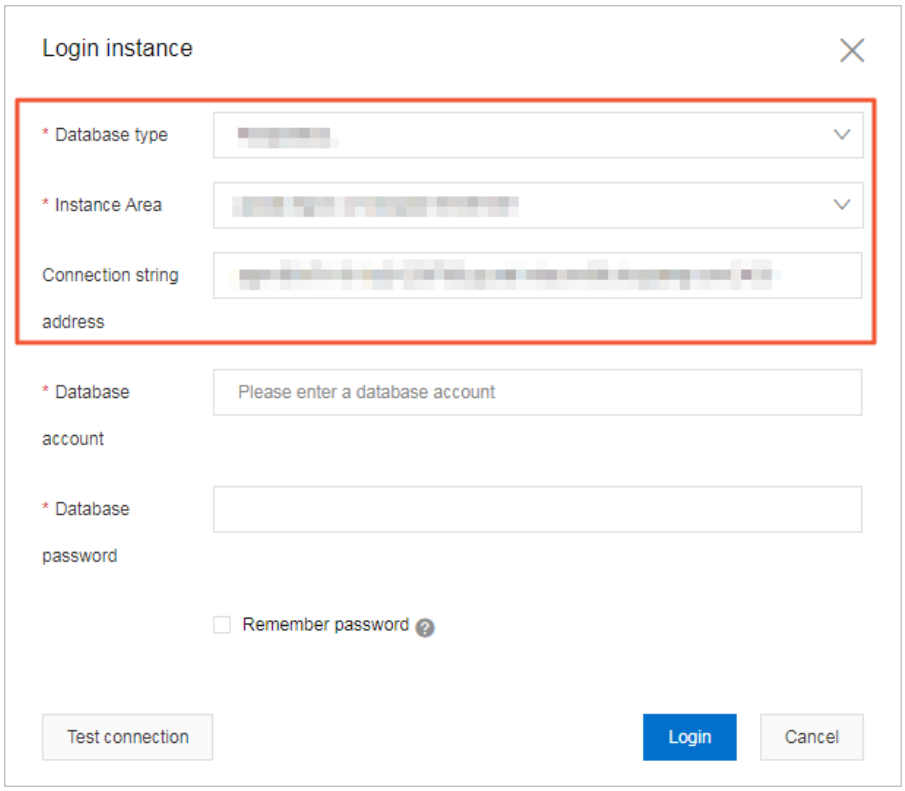
Before you start to use ApsaraDB RDS, you must create a database on an ApsaraDB RDS instance. This topic describes how to create a database on an ApsaraDB RDS for PostgreSQL instance.

Prerequisites

- An ApsaraDB RDS for PostgreSQL instance is created. For more information, see Create an instance.
- An account is created. For more information, see Create an account.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database password**, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**. If you want the browser to remember the password, select **Remember password** before you click **Login**.

   > **Note** If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see Configure an IP address whitelist.

7. The **SQLConsole** page appears after you log on to the instance. Execute a statement in the following format to create a database:

   ```
   CREATE DATABASE name;
   ```

   For example, if you want to create a database named test, execute the following statement:

   ```
   create database test;
   ```

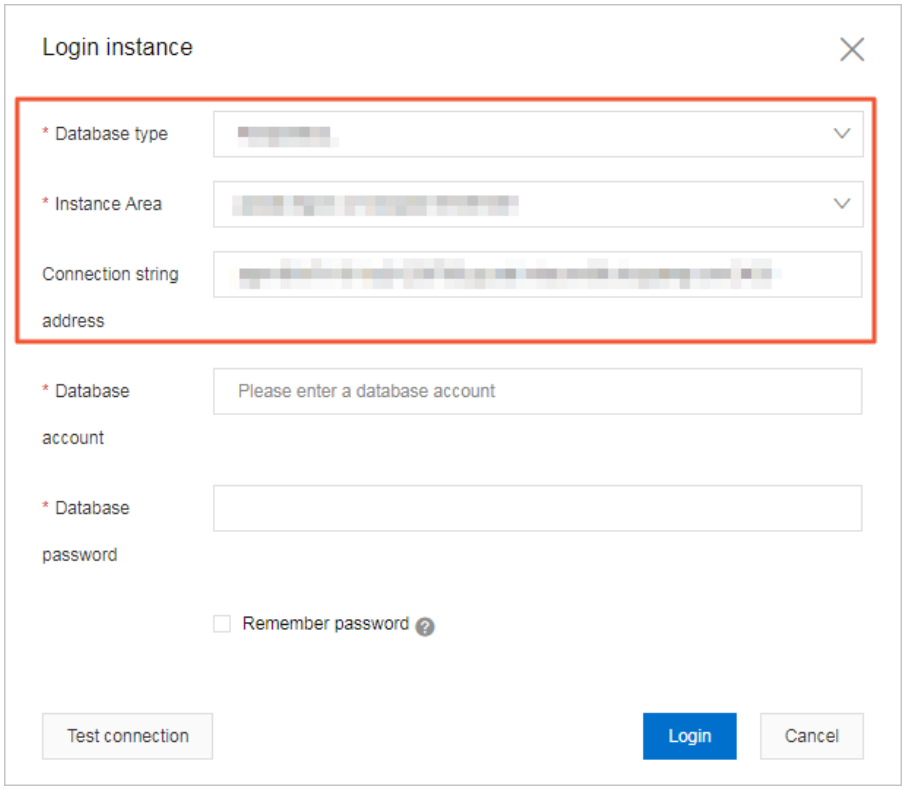
8. Click **execute**.

12.1.8.2. Delete a database

This topic describes how to delete a database in the ApsaraDB RDS for PostgreSQL console.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database password**, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**. If you want the browser to remember the password, select **Remember password** before you click **Login**.

   > **Note** If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see Configure an IP address whitelist.

7. The **SQLConsole** page appears after you log on to the instance. Execute the following statement to delete a database:

   ```
   drop database <database name>;
   ```

   For example, if you want to delete a database named test2, execute the following statement:

   ```
   drop database test2;
   ```

8. Click **execute**.

12.1.9. Networks, VPCs, and vSwitches

12.1.9.1. Change the network type of an ApsaraDB RDS for PostgreSQL instance

This topic describes how to change the network type of an ApsaraDB RDS for PostgreSQL instance between classic network and Virtual Private Cloud (VPC).

Prerequisites

ApsaraDB RDS instances use local SSDs.

Context

- Classic network: ApsaraDB RDS instances in the classic network are not isolated. You can block unauthorized access only by configuring IP address whitelists on these instances.
- VPC: Each VPC is an isolated network. We recommend that you use the VPC network type because it provides a higher security level.

  You can configure route tables, CIDR blocks, and gateways in a VPC. To smoothly migrate applications to the cloud, you can use leased lines or VPNs to connect your self-managed data center to a VPC to create a virtual data center.

Change the network type from VPC to classic network

Precautions

- The ApsaraDB RDS instance must be in a VPC.
- After you change the network type from VPC to classic network, the internal endpoint of the ApsaraDB RDS instance remains unchanged. However, the IP address that is associated with the internal endpoint changes.
- After you change the network type from VPC to classic network, you cannot connect Elastic Compute Service (ECS) instances deployed in VPCs to the ApsaraDB RDS instance by using the internal endpoint. You must update the endpoint for the applications deployed on the ECS instances.
- When you change the network type, a 30-second network interruption may occur. To avoid business interruption, change the network type during off-peak hours or make sure that your applications are configured with automatic reconnection policies.

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the Database Connection section, click **Switch to Classic Network**.
6. In the dialog box that appears, click **OK**.

   > **Note** After the network type is changed to classic network, only ECS instances within the classic network can connect to the ApsaraDB RDS instance by using the internal endpoint. You must configure the internal endpoint for the ECS instances.

7. Configure a whitelist to allow ECS instances within the classic network to connect to the ApsaraDB RDS instance by using the internal endpoint.

> **Note**
>
> - If the network isolation mode of the ApsaraDB RDS instance is standard whitelist, add the internal IP addresses of the ECS instances to a whitelist of your ApsaraDB RDS instance.
> - If the network isolation mode of the ApsaraDB RDS instance is enhanced whitelist, add the internal IP addresses of the ECS instances to a classic network whitelist. If no classic network whitelists are available, create a whitelist. For more information about the enhanced whitelist mode, see Switch to the enhanced whitelist mode.

Change the network type from classic network to VPC

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the Database Connection section, click **Switch to VPC**.
6. In the Switch to VPC dialog box, select a VPC and vSwitch and specify whether to retain the classic network endpoint.

> **Note**
>
> - Select a VPC. We recommend that you select the VPC where your ECS instances are deployed. Otherwise, the ECS instances cannot communicate with the ApsaraDB RDS instance over the internal network.
> - Select a vSwitch. If no vSwitches are available in the selected VPC, create one in the same zone where the ApsaraDB RDS instance is deployed. For more information, see Create a vSwitch in Quick Start of VPC User Guide.
> - Determine whether to select the **Reserve Original Classic Network Endpoint** option. The following table describes the details.
>
> - **Not selected**
>
> The classic network endpoint is not retained, and the endpoint of the instance changes to a VPC endpoint.
>
> When you change the network type, a 30-second network interruption may occur, and connections between ECS instances in the classic network and the ApsaraDB RDS instance are interrupted.
>
> - **Selected**
>
> The classic network endpoint is retained, and a new VPC endpoint is generated. In such cases, the ApsaraDB RDS instance runs in hybrid access mode. ECS instances in both the classic network and a VPC can connect to the ApsaraDB RDS instance over the internal network. You must set **Expiration Time (Important)** to **14 Days Later**, **30 Days Later**, **60 Days Later**, or **120 Days Later** for the classic network. You can also modify the expiration time after the network type is changed. For more information, see Hybrid network access mode.
>
> When you change the network type, no network interruptions occur. Connections between ECS instances in the classic network and the ApsaraDB RDS instance remain available until the classic network endpoint expires.
>
> To migrate your business to the VPC without interruption, you must add the VPC endpoint to access the ECS instances before the classic network endpoint expires. Seven days before the classic network endpoint expires, the system sends a text message to the phone number bound to your Apsara Stack account every day.
>
> For more information, see Hybrid access from both the classic network and VPCs.

7. Add the internal IP addresses of ECS instances in the selected VPC to a VPC whitelist. This allows the ECS instances to access the ApsaraDB RDS instance over the internal network. If no VPC whitelists are available, create a whitelist.

> **Note**
>
> - If you retain the classic network endpoint, add the VPC endpoint to the ECS instances before the classic network endpoint expires.
> - If you do not retain the classic network endpoint, connections between ECS instances in the classic network and the ApsaraDB RDS instance over the internal network are interrupted. You must add the new endpoint to ECS instances in the VPC immediately after the network type is changed.

12.1.9.2. Configure hybrid access from both the classic network and VPCs

This topic describes how to use the hybrid access solution of ApsaraDB RDS to change the network type of an instance from classic network to Virtual Private Cloud (VPC) without network interruptions.

Prerequisites

- The ApsaraDB RDS instance uses local SSDs.
- The ApsaraDB RDS instance is deployed in the classic network.
- Available VPCs and vSwitches exist in the zone where the ApsaraDB RDS instance is deployed.

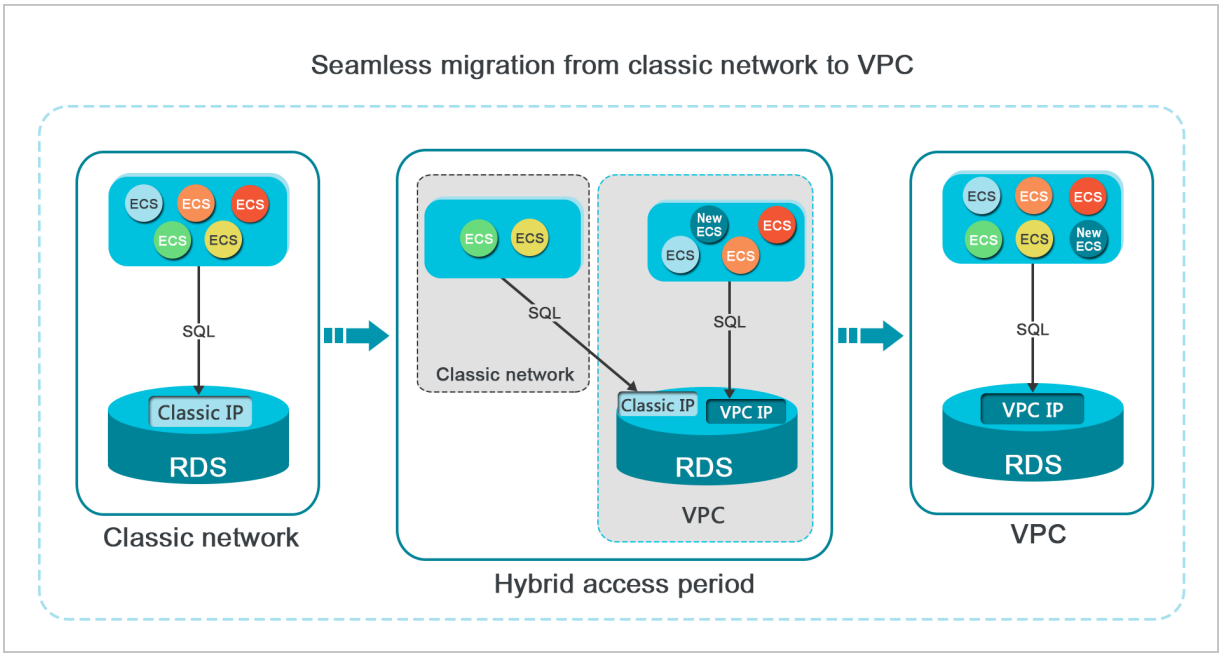
Context

In the past, when you change the network type of an ApsaraDB RDS instance from classic network to VPC, the internal endpoint of the ApsaraDB RDS instance would remain the same but the IP address bound to the endpoint would change to the corresponding IP address in the VPC. This change would cause a 30-second network interruption, and ECS instances within the classic network would not be able to access the ApsaraDB RDS instance by using the internal endpoint within this period. To smoothly change the network type, ApsaraDB RDS provides the hybrid access solution.

Hybrid access refers to the ability of an ApsaraDB RDS instance to be accessed by ECS instances in both the classic network and VPCs. During the hybrid access period, the ApsaraDB RDS instance reserves the original internal endpoint of the classic network and adds the internal endpoint of VPCs. This prevents network interruptions during the network type switchover.

For better security and performance, we recommend that you use the internal endpoint of VPCs. Hybrid access is available for a limited period of time. The internal endpoint of the classic network is released when the hybrid access period expires. In that case, your applications cannot access the ApsaraDB RDS database by using the internal endpoint of the classic network. You must configure the internal endpoint of VPCs in all your applications during the hybrid access period. This ensures smooth network switchover and minimizes the impact on your services.

For example, your company wants to use the hybrid access solution to change the network type from classic network to VPC. During the hybrid access period, some applications can access the database by using the internal endpoint of VPCs, and the other applications can access the database by using the original internal endpoint of the classic network. When all the applications access the database by using the internal endpoint of VPCs, the internal endpoint of the classic network can be released. The following figure illustrates the scenario.



Limits

During the hybrid access period, the instance has the following limits:

- The network type of your instance cannot be changed to classic network.
- Your instance cannot be migrated to another zone.

Change the network type from classic network to VPC

For more information, see Change the network type from classic network to VPC.

Change the expiration time for the original internal endpoint of the classic network

During the period in which your instance can be accessed over the classic network or VPCs, you can specify the expiration time for the endpoint of the classic network. The setting takes effect immediately. For example, if the endpoint of the classic network is about to expire on August 18, 2017 and you change the expiration time to 14 days later on August 15, 2017, the endpoint of the classic network is released on August 29, 2017.

To change the expiration time, perform the following steps:

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. On the **Instance Connection** tab, click **Change Expiration Time**.
6. In the **Change Expiration Time** dialog box, select an expiration time and click **OK**.

12.1.10. Monitoring

12.1.10.1. View monitored resources

ApsaraDB RDS provides a wide range of performance metrics. This topic describes how to view resource monitoring data in the ApsaraDB RDS console.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. On the **Monitoring** tab, select a time range to query the corresponding monitoring data. The following table lists the specific metrics.

Metric	Description
Disk Space	The used disk space of the instance. Unit: MB.
IOPS	The number of I/O requests of the data and log disks per second.
Memory Usage	The memory usage of the instance. Unit: %.
CPU Utilization	The CPU utilization of the instance. Unit: %.
Total Connections	The total number of current connections of the instance.

> **Note** You can click **Refresh** in the upper-right corner of the **Monitoring** tab to refresh the monitoring information.

12.1.10.2. Set a monitoring frequency

This topic describes how to set the monitoring frequency of an ApsaraDB RDS for PostgreSQL instance.

Context

ApsaraDB RDS for PostgreSQL provides the following monitoring frequencies:

- Every 5 seconds
- Every 60 seconds
- Every 300 seconds

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. On the **Monitoring** tab, click **Set Monitoring Frequency**.
6. In the **Set Monitoring Frequency** dialog box, select a monitoring frequency and click **OK**.

12.1.11. Data security

12.1.11.1. Switch to the enhanced whitelist mode

This topic describes how to switch from the standard whitelist mode to the enhanced whitelist mode for an ApsaraDB RDS instance. The enhanced whitelist mode provides higher security.

Network isolation modes

ApsaraDB RDS instances support the following network isolation modes:

- Standard whitelist mode

  IP addresses from both the classic network and VPCs are added to the same IP address whitelist. However, the standard whitelist mode may incur security risks. Therefore, we recommend that you switch the network isolation mode to enhanced whitelist.

- Enhanced whitelist mode

  IP addresses from the classic network and VPCs are added to different IP address whitelists. When you create an enhanced IP address whitelist, you must specify its network type.

Changes after you switch to the enhanced whitelist mode

- If the network type of the instance is VPC, the system generates a new whitelist that contains the same IP addresses as the original whitelists. The new IP address whitelist applies only to access from VPCs.
- If the network type of the instance is classic network, the system generates a new whitelist that contains the same IP addresses as the original whitelists. The new IP whitelist applies only to access from the classic network.
- If the instance supports access from both the classic network and VPCs, two new IP address whitelists are created, and each contains the same IP addresses as the original whitelists. One whitelist applies to access from VPCs, and the other applies to access from the classic network.

> **Note** After you switch to the enhanced whitelist mode, the configured ECS instance groups remain unchanged.

Precautions

- You can switch from the standard whitelist mode to the enhanced whitelist mode, but not the other way around.
- In enhanced whitelist mode, a classic network whitelist also allows access from the Internet. If you want to access the instance from a host over the Internet, you can add the public IP address of the host to a classic network whitelist.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Switch to Enhanced Whitelist (Recommended)**.
6. In the message that appears, click **Confirm**.

12.1.11.2. Configure an IP address whitelist

This topic describes how to configure a whitelist for an ApsaraDB RDS instance. Only entities that are listed in a whitelist can access your ApsaraDB RDS instance.

Context

Whitelists make your ApsaraDB RDS instance more secure and do not interrupt the operations of your ApsaraDB RDS instance when you configure whitelists. We recommend that you perform maintenance on your whitelists on a regular basis.

To configure a whitelist, perform the following operations:

- Configure a whitelist: Add IP addresses to allow them to connect to the ApsaraDB RDS instance.

  > **Note** The IP address whitelist labeled default contains only the default IP address 0.0.0.0/0, which allows all entities to access your ApsaraDB RDS instance.

- Configure an ECS security group: Add an ECS security group for the ApsaraDB RDS instance to allow ECS instances in the group to connect to the ApsaraDB RDS instance.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Edit** corresponding to the **default** whitelist.

   > **Note** You can also click **Create Whitelist** to create a whitelist.

6. In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks used to access the instance and click

OK. The following section describes the rules:

- If you enter the CIDR block 10.10.10.0/24 in the IP Addresses field, all IP addresses in the 10.10.10.X format can access your ApsaraDB RDS instance.
- If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
- If you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all ECS instances created within your Alibaba Cloud account are displayed. You can select the required IP addresses to add them to the IP address whitelist.

12.1.11.3. Configure SSL encryption

This topic describes how to enable SSL encryption for an ApsaraDB RDS instance.

Prerequisites

The ApsaraDB RDS instance uses standard SSDs.

Precautions

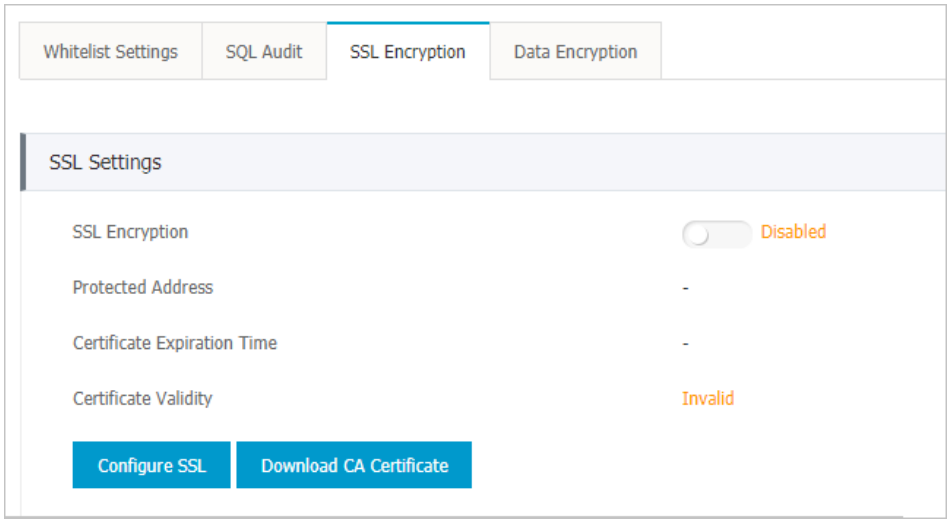
- After SSL encryption is enabled, data transmitted over an internal network or the Internet is encrypted by using SSL. SSL encryption protects data from leaks.
- After SSL encryption is enabled, you must close the existing connection and establish a new one to bring SSL encryption into effect.

Enable SSL encryption

SSL 3.0 has been upgraded by the Internet Engineering Task Force (IETF) to Transport Layer Security (TLS), but the term SSL encryption is still commonly used in the communications industry. Therefore, SSL encryption is used in this topic to refer to TLS encryption.

> **Note** ApsaraDB RDS supports TLS 1.0, 1.1, and 1.2.

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SSL Encryption** tab.

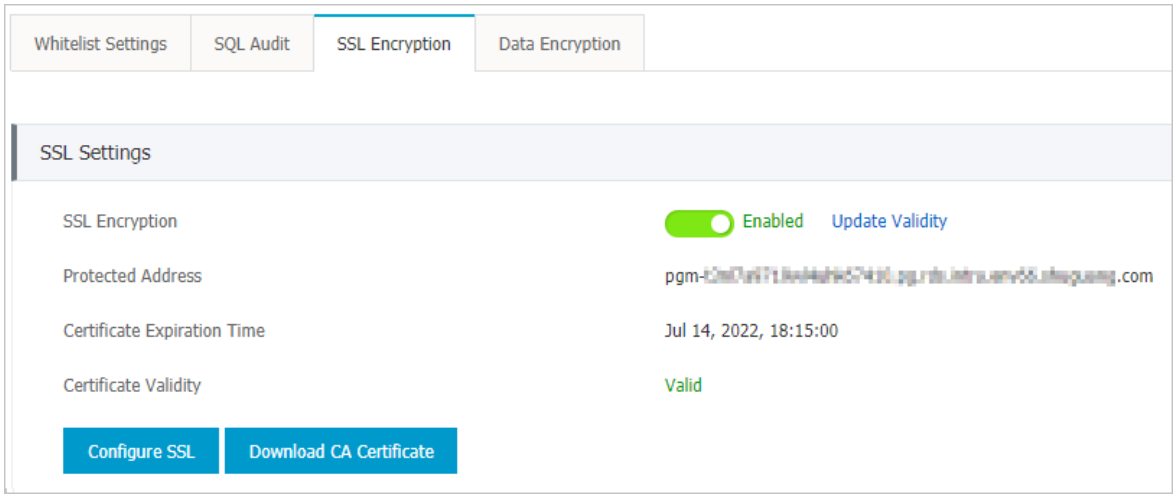


6. Click **Enable SSL**.

> **Note** After SSL encryption is enabled, you must set the SSL mode to **Prefer** when you log on from your client.

Disable SSL encryption

1. Log on to the ApsaraDB for RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SSL Encryption** tab.



6. Click **Disable SSL**.

12.1.11.4. Configure data encryption

This topic describes how to configure data encryption for an ApsaraDB RDS instance that uses standard or enhanced SSDs. The disk encryption feature maximizes the protection for your data and eliminates the need to modify business or application configurations. ApsaraDB RDS automatically applies disk encryption to both the snapshots that are generated from the encrypted SSDs and the SSDs that are created from those snapshots.

Prerequisites

The storage type of the instance is standard SSD.

Configure disk encryption

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

> **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
>
> - Uppercase or lowercase letters
> - Digits
> - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products** > Security > Key Management Service.
5. On the Keys page, click **Create Key**.
6. Configure the following parameters.

Section	Parameter	Description
Region	**Organization**	The organization to which the key belongs.
	Resource Set	The resource set to which the key belongs.
	Region	The region to which the key belongs.
Basic Settings	**Key Type**	KMS supports the following key types: ○ Symmetric keys: ■ Aliyun_AES_256 ■ Aliyun_SM4 ○ Asymmetric keys: ■ RSA_2048 ■ EC_P256 ■ EC_P256K ■ EC_SM2
	Key Purpose	ENCRYPT/DECRYPT: The purpose of the CMK is to encrypt or decrypt data.
	Protection Level	○ SOFTWARE: Use a software module to protect the CMK. ○ HSM: Host the CMK in a hardware security module (HSM). Managed HSM uses the HSM as dedicated hardware to safeguard the CMK.
	Alias	The identifier of the CMK. For more information, see Use aliases in *KMS User Guide*.
	Description	The description of the CMK.

Section	Parameter	Description
Advanced Settings	Rotation Period	Specifies whether to enable automatic rotation. If you choose to enable automatic rotation, you must select a rotation period. For more information about rotation, see Key rotation in *KMS User Guide*. Valid values: ○ 30 Days ○ 90 Days ○ 180 Days ○ 365 Days ○ Custom: Customize a period that ranges from 7 to 730 days. **Note** You can specify this parameter only if Key Type is set to Aliyun_AES_256 or Aliyun_SM4.
	Key Material Source	The source of key material. ○ Key Management Service: Use KMS to generate key material. ○ External: Manually import external key material. **Note** If **Rotation Period** is set to **Enable**, the **External** option is unavailable.

7. Click **Submit**.
8. Create an ApsaraDB RDS instance with disk encryption enabled. For more information, see Create an ApsaraDB RDS for PostgreSQL instance that uses standard or enhanced SSDs.

12.1.12. Logs and audit

12.1.12.1. Configure SQL audit

This topic describes how to configure the SQL audit feature to audit SQL executions and check the details. SQL audit does not affect instance performance.

Precautions

- SQL audit does not affect instance performance.
- SQL audit logs are retained for 30 days.
- Log files exported from SQL audit are retained for two days. The system deletes files that are retained for longer than two days.
- SQL audit is disabled by default. You must manually enable it.
- You cannot view logs that are generated before SQL audit is enabled.

Enable SQL audit

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.

5. Click the **SQL Audit** tab.
6. Click **Enable SQL Audit** or **Enable now**.
7. In the message that appears, click **Confirm**.

> **Note** After SQL audit is enabled, you can query SQL information based on conditions such as the time, database, user, and keyword.

Disable SQL audit

You can disable SQL audit when it is no longer needed. To disable SQL audit, perform the following steps:

> **Notice** After SQL audit is disabled, all SQL audit logs are deleted. We recommend that you export and store audit logs to your computer before you disable SQL audit.

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SQL Audit** tab. Click **Export File**.

> **Note** If more than 1 million SQL audit logs meet the filter conditions you specify, only 1 million logs can be exported. SQL audit logs are exported at a speed of 900 entries per second. It takes about 20 minutes to export 1 million SQL audit logs.

6. Click **Files**. Find a file and click Download in the **Action** column to download the file to your computer.
7. Click **Disable SQL Audit**.
8. In the message that appears, click **Confirm**.

12.1.12.2. Manage logs

You can view logs for errors, slow queries, and primary/secondary instance switching for ApsaraDB RDS for PostgreSQL instances in the ApsaraDB RDS console or by executing SQL statements. These logs help you troubleshoot errors. This topic describes how to manage logs in the console.

Procedure

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Logs**.
5. On the **Logs** page, click the **Error Logs**, **Slow Query Logs**, or **Primary/Secondary Switching Logs** tab, select a time range, and then click **Search**.

Tab	Description
Error Logs	Records database running errors that occurred within the last month.
Slow Query Logs	Records SQL statements within the last month that took longer than one second to execute. Duplicated SQL statements are removed.

Tab	Description
Primary/Secondary Switching Logs	Records switchovers between the primary and secondary instances within the last month.

12.1.13. Backup

12.1.13.1. Back up an ApsaraDB RDS for PostgreSQL instance

This topic describes how to back up an ApsaraDB RDS for PostgreSQL instance. You can configure a backup policy that is used to automatically back up your ApsaraDB RDS instance. If you do not configure a backup policy, the default backup policy is used. You can also manually back up your ApsaraDB RDS instance.

Precautions

- Do not perform data definition language (DDL) operations during a backup. If you do so, the backup may fail due to table locks.
- We recommend that you back up your ApsaraDB RDS instance during off-peak hours.
- If the amount of data is large, it may take a long time to back up your ApsaraDB RDS instance.
- Backup files are retained for a specified retention period. We recommend that you download the required backup files to your computer before they are deleted.

Backup description

ApsaraDB RDS for PostgreSQL allows you to perform full physical backup and back up archived redo log files of databases.

Configure a backup policy to automatically back up your ApsaraDB RDS instance

ApsaraDB RDS automatically backs up your instance based on the specified backup policy.

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. On the **Backup and Restoration** page, click the **Backup Settings** tab and click **Edit**.
6. In the dialog box that appears, configure the following parameters and click **OK**. The following table lists the parameters.

Parameter	Description
Data Retention Period	The number of days for which you want to retain data backup files. Valid values: 7 to 730. Unit: days. Default value: 7.
Backup Cycle	The cycle to create backups. You can select one or more days of the week. **Note** To ensure data security, we recommend that you back up your ApsaraDB RDS instance at least twice a week.
Backup Time	The period of time for which you want to back up data. Unit: hours.

Parameter	Description
Log Backup	Specifies whether to enable the log backup feature. **Notice** If you disable this feature, all log backup files are deleted and your instance cannot be restored to previous points in time.
Log Retention Period	◦ The period of time for which you want to retain log backup files. Valid values: 7 to 730. Unit: days. Default value: 7. ◦ The log retention period must be less than or equal to the data retention period.

Manually back up your ApsaraDB RDS instance

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Back Up Instance**. The **Back Up Instance** dialog box appears.
5. Select the backup mode and backup policy, and click **OK**.

> **Note** The backup mode is **Full Backup** and the backup policy is **Instance Backup**.

What's next

You can click the icon in the upper-right corner of the page to view the task progress displayed in the **Task Progress** list.

12.1.13.2. Download data and log backup files

This topic describes how to download unencrypted data and log backup files in the ApsaraDB RDS console to archive the files and restore data to an on-premises database.

Procedure

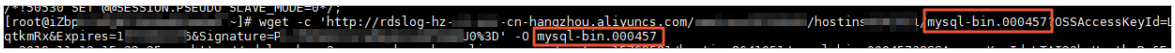
1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration** to go to the **Backup and Restoration** page.
5. Click the Data Backup or Archived Logs tab.
   - To download data backup files, click the **Data Backup** tab.
   - To download log files, click the **Archived Logs** tab.
6. Select a time range to which you want to restore the instance.
7. Find the data backup or log file that you want to download and click **Download** in the **Actions** column.

> **Note**
> - If you want to use a data backup file to restore data, select the backup file that is the closest to the time for restoration.
> - If you want to use a log file to restore data to an on-premises database, take note of the following items:
>   - The instance No. of the log file must be the same as that of the data backup file.
>   - The start time of the log file must be later than the data backup time and earlier than the time for restoration.

8. In the message that appears, select a download method.

Download method	Description
Download	Download the file by using the public endpoint.
Copy Internal Endpoint	Copy the internal endpoint to download the file. If your ECS and ApsaraDB RDS instances are deployed within the same region, you can log on to the ECS instance and use the internal endpoint to download the file. This method is fast and secure.
Copy Public Endpoint	Copy the public endpoint to download the file. If you want to use other tools to download the file, use the public endpoint.

> **Note** If you use a Linux operating system, you can run the following command to download the file:
>
> ```
> wget -c '<Public endpoint of the backup file, which is the download URL>' -O <File name>
> ```
>
> - The -c option enables resumable download.
> - The -O option saves the downloaded file by using a specified name. We recommend that you use the file name contained in the download URL.
> - If the URL contains more than one parameter, enclose the download URL in a pair of single quotation marks (').



12.1.13.3. Create a logical backup for an ApsaraDB RDS for PostgreSQL instance

This topic describes how to use pg_dump to create a logical backup for an ApsaraDB RDS for PostgreSQL instance and export the backup file to your computer.

Context

The pg_dump utility provided with PostgreSQL is used to back up individual databases. For more information, visit pg_dump.

In this example, an ApsaraDB RDS for PostgreSQL instance that runs Linux 7 and PostgreSQL 10 is used.

Prerequisites

- The IP address of your ECS instance or host is added to a whitelist of the ApsaraDB RDS for PostgreSQL instance. For more information, see Configure an IP address whitelist.
- Your ECS instance or host runs the same version of PostgreSQL as the ApsaraDB RDS for PostgreSQL instance.

Precautions

We recommend that you use the privileged account of the ApsaraDB RDS for PostgreSQL instance to ensure that you have all the required permissions.

Back up a database

1. Log on to your ECS instance or host. Then, run the following command to back up a database from the ApsaraDB RDS for PostgreSQL instance:

```
pg_dump -h '<hostname>' -U <username> -p <port> -Fc <dbname> > <dumpdir>
```

Parameter	Description
hostname	The endpoint of the ApsaraDB RDS for PostgreSQL instance. **Note** If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS and ApsaraDB RDS instances have the same network type. If both instances use the VPC network type, make sure that both instances are deployed within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
-Fc	The output file format. `-Fc` specifies the custom format, which is ideal when you use pg_restore to import logical backup files and restore databases. For more information, visit pg_dump.
dbname	The name of the database that you want to back up.
dumpdir	The directory and name of the logical backup file to export.

Example:

```
pg_dump -h 'pgm-bpxxxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -Fc testdb > /tmp/testdb.dump
```

2. When `Password:` appears, enter the password of the privileged account of the ApsaraDB RDS for PostgreSQL instance and press the Enter key.

Back up one or more tables

1. Log on to your ECS instance or host. Then, run the following command to back up one or more tables from a database in the ApsaraDB RDS for PostgreSQL instance:

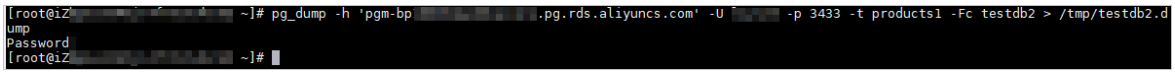
```
pg_dump -h '<hostname>' -U <username> -p <port> -t <table> -Fc <dbname> > <dumpdir>
```

Parameter	Description
hostname	The endpoint of the ApsaraDB RDS for PostgreSQL instance. **Note** If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS and ApsaraDB RDS instances have the same network type. If both instances use the VPC network type, make sure that both instances are deployed within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
table	The name of the table that you want to back up. You can use `-t <table>` to specify more than one table.
-Fc	The output file format. `-Fc` specifies the custom format, which is ideal when you use pg_restore to import logical backup files and restore databases. For more information, visit pg_dump.
dbname	The name of the database that you want to back up.
dumpdir	The directory and name of the logical backup file to export.

Example:

```
pg_dump -h 'pgm-bpxxxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -t products1 -Fc testdb2 > /tmp/testdb2.dump
```

2. When `Password:` appears, enter the password of the privileged account of the ApsaraDB RDS for PostgreSQL instance and press the Enter key.



Back up a database with one or more tables excluded

1. Log on to your ECS instance or host. Then, run the following command to back up a database from the ApsaraDB RDS instance with one or more tables excluded:

```
pg_dump -h '<hostname>' -U <username> -p <port> -T <table> -Fc <dbname> > <dumpdir>
```

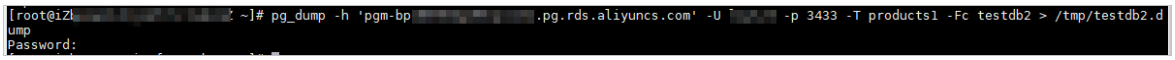
Parameter	Description

Parameter	Description
hostname	The endpoint of the ApsaraDB RDS for PostgreSQL instance. **Note** If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS and ApsaraDB RDS instances have the same network type. If both instances use the VPC network type, make sure that both instances are deployed within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
table	The name of the table that you want to exclude. You can use `-T <table>` to specify more than one table.
-Fc	The output file format. `-Fc` specifies the custom format, which is ideal when you use pg_restore to import logical backup files and restore databases. For more information, visit pg_dump.
dbname	The name of the database that you want to back up.
dumpdir	The directory and name of the logical backup file to export.

Example:

```
pg_dump -h 'pgm-bpxxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -T products1 -Fc testdb2 > /tmp/testdb2.dump
```

2. When `Password:` appears, enter the password of the privileged account of the ApsaraDB RDS for PostgreSQL instance and press the Enter key.



Back up the schema of a database with data excluded

1. Log on to your ECS instance or host. Then, run the following command to back up the schema of a database from the ApsaraDB RDS for PostgreSQL instance:

```
pg_dump -h '<hostname>' -U <username> -p <port> -s -Fc <dbname> > <dumpdir>
```

Parameter	Description

Parameter	Description
hostname	The endpoint of the ApsaraDB RDS for PostgreSQL instance. **Note** If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS and ApsaraDB RDS instances have the same network type. If both instances use the VPC network type, make sure that both instances are deployed within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
-s	Specifies whether to back up only the schema of the database. The data of the database is not backed up. For more information, visit pg_dump.
-Fc	The output file format. `-Fc` specifies the custom format, which is ideal when you use pg_restore to import logical backup files and restore databases. For more information, visit pg_dump.
dbname	The name of the database that you want to back up.
dumpdir	The directory and name of the logical backup file to export.

Example:

```
pg_dump -h 'pgm-bpxxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -s -Fc testdb2 > /tmp/testdb2.dump
```

2. When `Password:` appears, enter the password of the privileged account of the ApsaraDB RDS for PostgreSQL instance and press the Enter key.

12.1.13.4. Create a full backup of an ApsaraDB RDS for PostgreSQL instance

This topic describes how to use the pg_basebackup utility provided by open source PostgreSQL to create a full backup of your ApsaraDB RDS for PostgreSQL instance and export the backup files to your computer.

Prerequisites

- The IP address of your ECS instance or host is added to a whitelist of your ApsaraDB RDS for PostgreSQL instance. For more information, see Configure an IP address whitelist.
- Your ECS instance or host runs the same version of PostgreSQL as the ApsaraDB RDS for PostgreSQL instance.

Context

pg_basebackup backs up all data of a PostgreSQL instance. Backup files can be used for point-in-time recovery.

For more information, visit pg_basebackup.

In this example, CentOS 7 is used to create a full backup.

Precautions

We recommend that you use the privileged account of the ApsaraDB RDS for PostgreSQL instance to ensure that you have all the required permissions.

Procedure

> **Note** pg_basebackup cannot back up a single database or database object. For more information about how to back up a single database or database object, see Create a logical backup for an ApsaraDB RDS for PostgreSQL instance.

1. Log on to your ECS instance or host. Then, run the following command to back up a database from the ApsaraDB RDS for PostgreSQL instance:

   ```
   pg_basebackup -Ft -Pv -Xf -z -D <backupdir> -Z5 -h '<hostname>' -p <port> -U <username> -W
   ```

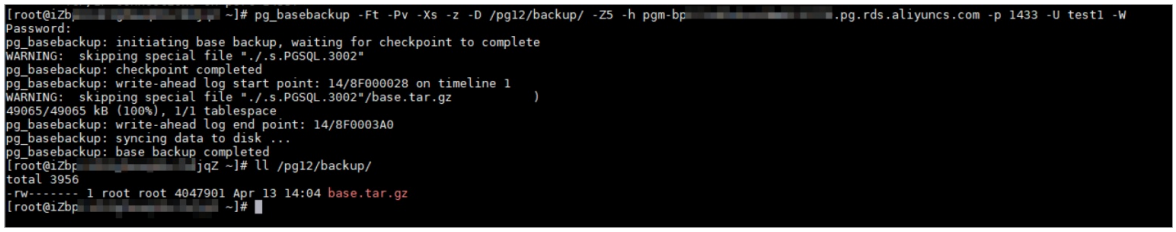
   The following table describes the parameters in this command. For more information, visit pg_basebackup.

Parameter	Description
backupdir	The directory of backup files that are exported. The system automatically creates this directory. However, if this directory already exists and is not empty, the system reports an error.
hostname	The internal endpoint of the ApsaraDB RDS for PostgreSQL instance. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
username	A username of the ApsaraDB RDS for PostgreSQL instance.

   Example:

   ```
   pg_basebackup -Ft -Pv -Xf -z -D /pg12/backup1/ -Z5 -h pgm-bpxxxxx.pg.rds.aliyuncs.com -p 1433 -U test1 -W
   ```

2. When `Password:` appears, enter the password of the username of the ApsaraDB RDS for PostgreSQL instance and press the Enter key.

12.1.14. Restoration

12.1.14.1. Restore data of an ApsaraDB RDS for PostgreSQL instance

This topic describes how to use the backup data of an ApsaraDB RDS for PostgreSQL instance to restore data.

Precautions

- The new instance must have the same whitelist, backup, and parameter settings as the original instance.
- The new instance must have the same data and account information as the backup set or instance at the time point.

Prerequisites

The original instance must meet the following requirements:

- The original instance is in the Running state and is not locked.
- The original instance does not have ongoing migration tasks.
- If you want to restore data to a point in time, the log backup feature is enabled for the original instance.
- If you want to restore an instance from a backup set, the original instance has at least one backup set.

Restore data of an ApsaraDB RDS for PostgreSQL instance

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. In the upper-right corner of the page, click **Restore Database (Previously Clone Database)**.
6. Configure the following parameters.

Section	Parameter	Description
Region	**Region**	The region where the instance is deployed.
Restore Database	**Restore Mode**	◦ **By Time**: You can restore data to a point in time within the retention period of the log backup. For more information about how to view or change the retention period of log backups, see Back up an ApsaraDB RDS for PostgreSQL instance. ◦ **By Backup Set** **Note** The **By Time** option appears only when the log backup feature is enabled.
	Restore Time	The time to which the database is restored. This parameter is displayed when you set **Restore Mode** to **By Time**.
	Backup Set	The backup set used to restore the database. This parameter is displayed when you set **Restore Mode** to **By Backup Set**.
	Instance Name	The name of the instance.
	Database Engine	The engine of the database. The value of this parameter is set to **PostgreSQL** and cannot be changed.
	Engine Version	The version of the database engine. The value of this parameter is set to the engine version of the current instance and cannot be changed.

Section	Parameter	Description
Specifications	**Edition**	The edition of the instance.
	Storage Type	The storage type of the instance. The value of this parameter is set to the storage type of the current instance and cannot be changed.
	Instance Type	The instance type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. For more information, see Instance types in *Instance types* of *ApsaraDB RDS Product Introduction*.
	Storage Capacity	The storage capacity of the instance, including the space to store data, system files, binlog files, and transaction files. Valid values: 20 to 600. Unit: GB. The value must be in 1 GB increments.
Network Type	**Network Type**	The network type of the instance. ApsaraDB RDS instances support the following network types: ○ **Classic Network**: Cloud services on a classic network are not isolated from each other. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ○ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for improved security. **Note** If you set the network type to **VPC**, you must also select a **VPC** and a **vSwitch**.

7. Click **Submit**.

12.1.14.2. Restore data from a logical backup file

This topic describes how to restore data from a logical backup file to an ApsaraDB RDS for PostgreSQL instance or an on-premises PostgreSQL database.

Context

A logical backup file is used to restore a small amount of data, such as data in a table. For a large amount of data, we recommend that you restore it from a full physical backup file to a new ApsaraDB RDS instance and then use Data Transmission Service (DTS) to migrate data to the original ApsaraDB RDS instance.

Prerequisites

Data in the ApsaraDB RDS for PostgreSQL instance is logically backed up. For more information, see Create a logical backup for an ApsaraDB RDS for PostgreSQL instance.

Precautions

- We recommend that you do not restore data to the default postgres database.
- When you restore the data of a table, the system does not restore the database objects on which the table depends. The restoration may fail.

Restore the data of a database

1. Log on to the ECS instance or on-premises host that houses the logical backup file and run the following command to restore the data of a database:

```
pg_restore -h '<hostname>' -U <username> -p <port> -d <dbname> <dumpdir>
```

Parameter	Description
hostname	The endpoint of the ApsaraDB RDS for PostgreSQL instance. **Note** If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS and ApsaraDB RDS instances have the same network type. If both instances are of the VPC network type, make sure that both instances reside within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
dbname	The name of the database whose data you want to restore.
dumpdir	The directory and name of the logical backup file to use.

Example:

```
pg_restore -h 'pgm-bpxxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -d testdb2 /tmp/testdb.dump
```

2. When `Password:` appears, enter the password of the privileged account of your ApsaraDB RDS instance and press the Enter key.

Note You can ignore alerts generated by the embedded plpgsql plug-in.

```
[root@iZb ~]# pg_restore -h 'pgm-bp .pg.rds.aliyuncs.com' -U  -p 3433 -d testdb4 /tmp/testdb2.dump
Password:
pg_restore: [archiver (db)] Error while PROCESSING TOC:
pg_restore: [archiver (db)] Error from TOC entry 3076; 0 0 COMMENT EXTENSION plpgsql
pg_restore: [archiver (db)] could not execute query: ERROR:  must be owner of extension plpgsql
    Command was: COMMENT ON EXTENSION plpgsql IS 'PL/pgSQL procedural language';


WARNING: errors ignored on restore: 1
```

Restore the data of a table

1. Log on to the ECS instance or on-premises host that houses the logical backup file and run the following command to restore the data of a table:

```
pg_restore -h '<hostname>' -U <username> -p <port> -d <dbname> -t <table> -c <dumpdir>
```

Parameter	Description

Parameter	Description
hostname	The endpoint of the ApsaraDB RDS for PostgreSQL instance. **Note** If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS and ApsaraDB RDS instances have the same network type. If both instances are of the VPC network type, make sure that both instances reside within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
dbname	The name of the database whose data you want to restore.
table	The name of the table whose data you want to restore.
-c	`-c`: specifies to delete the database objects on which the table depends before data restoration. For more information, visit pg_restore.
dumpdir	The directory and name of the logical backup file to use.

Example:

```
pg_restore -h 'pgm-bpxxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -d testdb2 -t products -c /tmp
/testdb.dump
```

2. When `Password:` appears, enter the password of the privileged account of your ApsaraDB RDS instance and press the Enter key.

Restore the schema of a database with data excluded

1. Log on to the ECS instance or on-premises host that houses the logical backup file and run the following command to restore only the schema of a database:

```
pg_restore -h '<hostname>' -U <username> -p <port> -d <dbname> -s <dumpdir>
```

Parameter	Description

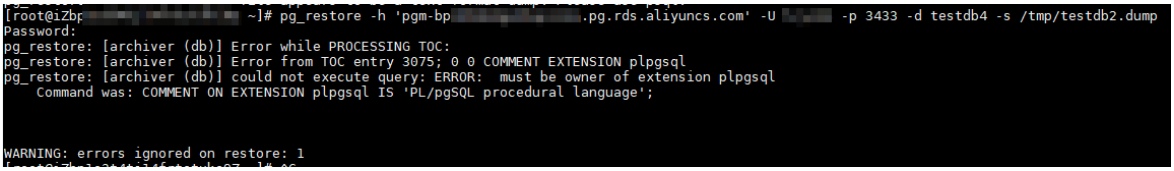
Parameter	Description
hostname	The endpoint of the ApsaraDB RDS for PostgreSQL instance. **Note** If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS and ApsaraDB RDS instances have the same network type. If both instances are of the VPC network type, make sure that both instances reside within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
dbname	The name of the database whose schema you want to restore.
-s	`-s` : specifies to restore only the schema of the database. The data of the database is not restored. For more information, visit pg_restore.
dumpdir	The directory and name of the logical backup file to use.

Example:

```
pg_restore -h 'pgm-bpxxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -d testdb4 -s /tmp/testdb2.dump
```

2. When `Password:` appears, enter the password of the privileged account of your ApsaraDB RDS instance and press the Enter key.

Note You can ignore alerts generated by the embedded plpgsql plug-in.



12.1.15. Plug-ins

12.1.15.1. Plug-ins supported

This topic describes the plug-ins that are supported by ApsaraDB RDS for PostgreSQL and their available versions.

PostgreSQL 12

Plug-in	Version
btree_gin	1.3
btree_gist	1.5

Plug-in	Version
citext	1.6
cube	1.4
dblink	1.2
dict_int	1
earthdistance	1.1
fuzzystrmatch	1.1
hstore	1.6
intagg	1.1
intarray	1.2
isn	1.2
ltree	1.1
pg_buffercache	1.3
pg_prewarm	1.2
pg_stat_statements	1.7
pg_trgm	1.4
pgcrypto	1.3
pgrowlocks	1.2
pgstattuple	1.5
postgres_fdw	1
sslinfo	1.2
tablefunc	1
unaccent	1.1
plpgsql	1
plperl	1
pg_roaringbitmap	0.5.0
rdkit	3.8
mysql_fdw	1.1
ganos_geometry_sfcgal	3.0
ganos_geometry_topology	3.0

Plug-in	Version
ganos_geometry	3.0
ganos_networking	3.0
ganos_pointcloud_geometry	3.0
ganos_pointcloud	3.0
ganos_raster	3.0
ganos_spatialref	3.0
ganos_trajectory	3.0
ganos_tiger_geocoder	3.0
ganos_address_standardizer	3.0
ganos_address_standardizer_data_us	3.0
wal2json	2.0
hll	2.14
plproxy	2.9.0
tsm_system_rows	1.0
tsm_system_time	1.0
smlar	1.0
tds_fdw	1.0
bigm	1.2
timescaledb	1.7.1

PostgreSQL 11

Plug-in	Version
plpgsql	1
pg_stat_statements	1.6
btree_gin	1.3
btree_gist	1.5
citext	1.5
cube	1.4
rum	1.3
dblink	1.2

Plug-in	Version
dict_int	1
earthdistance	1.1
hstore	1.5
intagg	1.1
intarray	1.2
isn	1.2
ltree	1.1
pgcrypto	1.3
pgrowlocks	1.2
pg_prewarm	1.2
pg_trgm	1.4
postgres_fdw	1
sslinfo	1.2
tablefunc	1
timescaledb	1.7.1
unaccent	1.1
fuzzystrmatch	1.1
pgstattuple	1.5
pg_buffercache	1.3
zhparser	1
pg_pathman	1.5
plperl	1
orafce	3.8
pg_concurrency_control	1
varbitx	1
postgis	2.5.1
pgrouting	2.6.2
postgis_sfcgal	2.5.1
postgis_topology	2.5.1

Plug-in	Version
address_standardizer	2.5.1
address_standardizer_data_us	2.5.1
ogr_fdw	1
ganos_pointcloud	3.0
ganos_spatialref	3.0
log_fdw	1.0
wal2json	2.2
PL/v8	2.3.13
pg_cron	1.1
pase	0.0.1
hll	2.14
oss_fdw	1.1
tds_fdw	2.0.1
plproxy	2.9.0
tsm_system_rows	1.0
tsm_system_time	1.0
smlar	1.0
zombodb	4.0
bigm	1.2

PostgreSQL 10

Plug-in	Version
pg_stat_statements	1.6
btree_gin	1.2
btree_gist	1.5
chkpass	1
citext	1.4
cube	1.2
dblink	1.2
dict_int	1

Plug-in	Version
earthdistance	1.1
hstore	1.4
intagg	1.1
intarray	1.2
isn	1.1
ltree	1.1
pgcrypto	1.3
pgrowlocks	1.2
pg_prewarm	1.1
pg_trgm	1.3
postgres_fdw	1
sslinfo	1.2
tablefunc	1
unaccent	1.1
postgis_sfcgal	2.5.1
postgis_topology	2.5.1
fuzzystrmatch	1.1
postgis_tiger_geocoder	2.5.1
address_standardizer	2.5.1
address_standardizer_data_us	2.5.1
ogr_fdw	1
plperl	1
plv8	1.4.2
plls	1.4.2
plcoffee	1.4.2
uuid-ossp	1.1
zhparser	1
pgrouting	2.6.2
pg_hint_plan	1.3.0

Plug-in	Version
pgstattuple	1.5
oss_fdw	1.1
ali_decoding	0.0.1
varbitx	1
pg_buffercache	1.3
q3c	1.5.0
pg_sphere	1
smlar	1
rum	1.3
pg_pathman	1.5
aggs_for_arrays	1.3.1
mysql_fdw	1
orafce	3.6
plproxy	2.8.0
pg_concurrency_control	1
postgis	2.5.1
ganos_geometry_sfcgal	2.2
ganos_geometry_topology	2.2
ganos_geometry	2.2
ganos_networking	2.2
ganos_pointcloud_geometry	2.2
ganos_pointcloud	2.2
ganos_raster	2.2
ganos_spatialref	2.2
ganos_trajectory	2.2
ganos_tiger_geocoder	2.2
ganos_address_standardizer	2.2
ganos_address_standardizer_data_us	2.2

PostgreSQL 9.4

Plug-in	Version
plpgsql	1
pg_stat_statements	1.2
btree_gin	1
btree_gist	1
chkpass	1
citext	1
cube	1
dblink	1.1
dict_int	1
earthdistance	1
hstore	1.3
intagg	1
intarray	1
isn	1
ltree	1
pgcrypto	1.1
pgrowlocks	1.1
pg_prewarm	1
pg_trgm	1.1
postgres_fdw	1
sslinfo	1
tablefunc	1
tsearch2	1
unaccent	1
postgis	2.2.8
postgis_topology	2.2.8
fuzzystrmatch	1
postgis_tiger_geocoder	2.2.8
plperl	1

Plug-in	Version
pltcl	1
plv8	1.4.2
plls	1.4.2
plcoffee	1.4.2
uuid-ossp	1
zhparser	1
pgrouting	2.0.0
rdkit	3.4
pg_hint_plan	1.1.3
pgstattuple	1.2
oss_fdw	1.1
jsonbx	1
ali_decoding	0.0.1
varbitx	1
pg_buffercache	1
smlar	1
pg_sphere	1
q3c	1.5.0
pg_awr	1
imgsmlr	1
orafce	3.6
pg_concurrency_control	1

12.1.15.2. Use mysql_fdw to read data from and write data to a MySQL database

This topic describes how to use the mysql_fdw plug-in of ApsaraDB RDS for PostgreSQL to read data from and write data to a database on an ApsaraDB RDS for MySQL instance or a self-managed MySQL database.

Prerequisites

- The instance runs PostgreSQL 10.
- Communication between your ApsaraDB RDS for PostgreSQL instance and the MySQL database is normal.

Context

PostgreSQL 9.6 and later support parallel computing. PostgreSQL 11 can use joins on up to a billion data records to complete queries in seconds. A number of users prefer to use PostgreSQL to build small-sized data warehouses and process highly concurrent access requests. PostgreSQL 13 is under development. It will support columnar storage engines that further improve analysis capabilities.

The mysql_fdw plug-in establishes a connection to synchronize data from a MySQL database to your ApsaraDB RDS for PostgreSQL instance.

Procedure

1. Log on to a database of your ApsaraDB RDS for PostgreSQL instance. For more information, see Connect to an ApsaraDB RDS for PostgreSQL instance.
2. Create the mysql_fdw plug-in.

   ```
   create extension mysql_fdw;
   ```

3. Define a MySQL server.

   ```
   CREATE SERVER <Name of the MySQL server>
   FOREIGN DATA WRAPPER mysql_fdw
   OPTIONS (host '<Endpoint used to connect to the MySQL server>', port '<Port used to connect to th
   e MySQL server>');
   ```

   Example:

   ```
   CREATE SERVER mysql_server
   FOREIGN DATA WRAPPER mysql_fdw
   OPTIONS (host 'rm-xxx.mysql.rds.aliyuncs.com', port '3306');
   ```

4. Map the MySQL server to an account created on your ApsaraDB RDS for PostgreSQL instance. Then, the account can be used to access data in the MySQL database on the MySQL server.

   ```
   CREATE USER MAPPING FOR <Username of the account to which the MySQL server is mapped>
   SERVER <Name of the MySQL server>
   OPTIONS (username '<Username used to log on to the MySQL database>', password '<Password used to
   log on to the MySQL database>');
   ```

   Example:

   ```
   CREATE USER MAPPING FOR pgtest
   SERVER mysql_server
   OPTIONS (username 'mysqltest', password 'Test1234!') ;
   ```

5. Create a foreign MySQL table by using the account that you mapped to the MySQL server in the previous step.

   > **Note** The field names in the foreign MySQL table must be the same as those in the table of the MySQL database. You can choose to create only the fields you want to query. For example, if the table in the MySQL database contains the ID, NAME, and AGE fields, you can create only the ID and NAME fields in the foreign MySQL table.

   ```
   CREATE FOREIGN TABLE <Name of the foreign MySQL table> (<Name of Field 1> <Data type of Field 1>,
   <Name of Field 2> <Data type of Field 2>...) server <Name of the MySQL server> options (dbname '<
   Name of the MySQL database>', table_name '<Name of the table in the MySQL database>');
   ```

   Example:

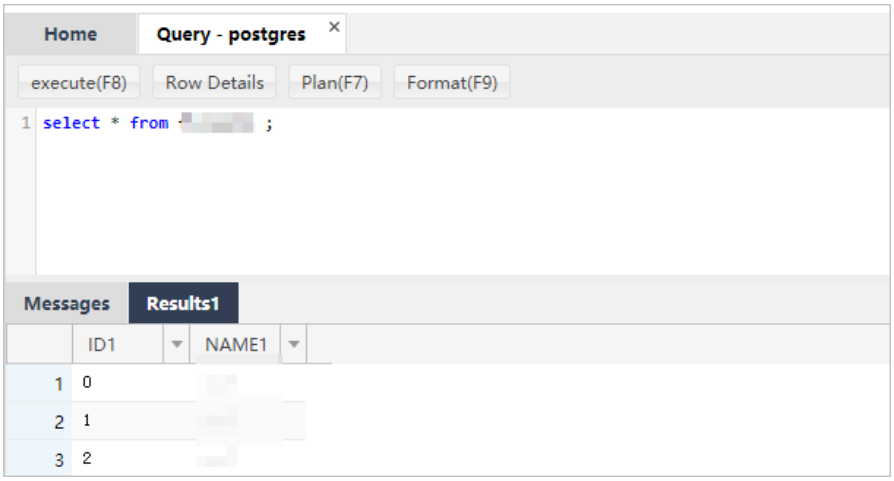
   ```
   CREATE FOREIGN TABLE ft_test (id1 int, name1 text) server mysql_server options (dbname 'test123',
   table_name 'test');
   ```

What to do next

You can use the foreign MySQL table to test the performance of read and write operations on the MySQL database.

> **Note** Data can be written to the table in the MySQL database only when the table is assigned a primary key. If the table is not assigned a primary key, the following error is reported:
>
> ```
> ERROR:  first column of remote table must be unique for INSERT/UPDATE/DELETE operation.
> ```

```
select * from ft_test ;
insert into ft_test values (2,'abc');
insert into ft_test select generate_series(3,100),'abc';
select count(*) from ft_test ;
```



Run `postgres=> explain verbose select count(*) from ft_test;` to find out how the requests sent from your ApsaraDB RDS for PostgreSQL instance are executed to query data from the MySQL database. Command output:

```
                                QUERY PLAN
---------------------------------------------------------------------------
 Aggregate  (cost=1027.50..1027.51 rows=1 width=8)
   Output: count(*)
   ->  Foreign Scan on public.ft_test  (cost=25.00..1025.00 rows=1000 width=0)
         Output: id, info
         Remote server startup cost: 25
         Remote query: SELECT NULL FROM `test123`.`test`
(6 rows)
```

12.1.15.3. Use oss_fdw to read and write foreign data files

This topic describes how to use the oss_fdw plug-in to load data between Object Storage Service (OSS) and PostgreSQL or PPAS databases.

oss_fdw parameters

The oss_fdw plug-in uses a method similar to other Foreign Data Wrapper (FDW) interfaces to encapsulate foreign data stored in OSS. You can use oss_fdw to read data stored in OSS. This process is similar to reading data tables. oss_fdw provides unique parameters to connect and parse file data in OSS.

> **Note**
> - oss_fdw can read and write files of the following types in OSS: TXT and CSV files as well as GZIP-compressed TXT and CSV files.
> - The value of each parameter must be enclosed in double quotation marks (") and cannot contain unnecessary spaces.

CREATE SERVER parameters

- ossendpoint: the endpoint used to access OSS over the internal network, also known as the host.
- id oss: the AccessKey ID of the OSS account.
- key oss: the AccessKey secret of the OSS account.
- bucket: the bucket where the data you want to access is stored. You must create an OSS account before you specify this parameter.

The following fault tolerance parameters can be used for data import and export. If network connectivity is poor, you can adjust these parameters to ensure successful import and export.

- oss_connect_timeout: the connection timeout period. Default value: 10. Unit: seconds.
- oss_dns_cache_timeout: the DNS timeout period. Default value: 60. Unit: seconds.
- oss_speed_limit: the minimum data transmission rate. Default value: 1024. Unit: bytes/s.
- oss_speed_time: the maximum waiting period during which the data transmission rate is lower than the minimum value. Default value: 15. Unit: seconds.

If the default values of oss_speed_limit and oss_speed_time are used, a timeout error occurs when the transmission rate is lower than 1,024 bytes/s for 15 consecutive seconds.

CREATE FOREIGN TABLE parameters

- filepath: a file name that contains a path in OSS.
  - The file name specified by this parameter contains the directory name but not the bucket name.
  - This parameter matches multiple files in the corresponding path in OSS. You can load multiple files to a database.
  - You can import files that adhere to the filepath or filepath.x format to a database. The values of x must be consecutive numbers starting from 1.

    For example, among the files named filepath, filepath.1, filepath.2, filepath.3, and filepath.5, the first four files are matched and imported. The filepath.5 file is not imported.
- dir: the virtual file directory in OSS.
  - The specified directory must end with a forward slash (/).
  - All files (excluding subfolders and files in subfolders) in the virtual file directory specified by dir are matched and imported to a database.
- prefix: the prefix of the path name corresponding to the data file. The prefix does not support regular expressions. The prefix, filepath, and dir parameters are mutually exclusive. Therefore, only one of them can be specified at a time.
- format: the file format, which can only be CSV.
- encoding: the file data encoding format. It supports common PostgreSQL encoding formats, such as UTF-8.
- parse_errors: the fault-tolerant parsing mode. If an error occurs during the parsing process, the entire row of data is ignored.
- delimiter: the string used to delimit columns.
- quote: the quote character for files.

- escape: the escape character for files.
- null: sets the column matching the specified string to null. For example, null 'test' is used to set the value of the 'test' column to null.
- force_not_null: sets the value of a column to a non-null value. For example, force_not_null 'id' is used to set the value of the 'id' column to empty strings.
- compressiontype: the format of the files to be read or written in OSS.
  - none: The files are uncompressed. This is the default value.
  - gzip: The files are compressed in the GZIP format.
- compressionlevel: the degree to which data files written to OSS are compressed. Valid values: 1 to 9. Default value: 6.

> **Note**
> - You must specify filepath and dir in the OPTIONS parameter.
> - You must specify filepath or dir.
> - The export mode can only be dir.

Export mode parameters for CREATE FOREIGN TABLE

- oss_flush_block_size: the buffer size for the data written to OSS at a time. Default value: 32. Valid values: 1 to 128. Unit: MB.
- oss_file_max_size: the maximum size of a data file allowed to be written to OSS. If a data file reaches the maximum size, the remaining data is written to another data file. Default value: 1024. Valid values: 8 to 4000. Unit: MB.
- num_parallel_worker: the maximum number of threads that are allowed to run in parallel to compress the data written to OSS. Valid values: 1 to 8. Default value: 3.

Auxiliary functions

FUNCTION oss_fdw_list_file (relname text, schema text DEFAULT 'public')

- This function obtains the name and size of the OSS file that a foreign table matches.
- The file size is measured in bytes.

The following result is returned after `select * from oss_fdw_list_file('t_oss');` is executed:

```
              name              |   size
--------------------------------+-----------
 oss_test/test.gz.1  | 739698350
 oss_test/test.gz.2  | 739413041
 oss_test/test.gz.3  | 739562048
(3 rows)
```

Auxiliary features

oss_fdw.rds_read_one_file: In read mode, this feature is used to specify a file to match the foreign table. The foreign table matches only the specified file during data import.

Example: set oss_fdw.rds_read_one_file = 'oss_test/example16.csv.1';

The following result is returned after `set oss_fdw.rds_read_one_file = 'oss_test/test.gz.2';` and `select * from oss_fdw_list_file('t_oss');` are executed:

```
             name              |   size
-------------------------------+-----------
  oss_test/test.gz.2  | 739413041
(1 rows)
```

oss_fdw example

```
# Create the plug-in for a PostgreSQL database.
create extension oss_fdw; -- For a PPAS database, execute select rds_manage_extension('create','oss_fdw');
# Create a server.
CREATE SERVER ossserver FOREIGN DATA WRAPPER oss_fdw OPTIONS
     (host 'oss-cn-hangzhou.aliyuncs.com', id 'xxx', key 'xxx', bucket 'mybucket');
# Create an OSS foreign table.
CREATE FOREIGN TABLE ossexample
    (date text, time text, open float,
     high float, low float, volume int)
     SERVER ossserver
     OPTIONS ( filepath 'osstest/example.csv', delimiter ',' ,
         format 'csv', encoding 'utf8', PARSE_ERRORS '100');
# Create a table named example to which to import data.
create table example
        (date text, time text, open float,
         high float, low float, volume int);
# Load data from ossexample to example.
insert into example select * from ossexample;
# Result
# oss_fdw estimates the file size in OSS and formulates a query plan.
explain insert into example select * from ossexample;
                            QUERY PLAN
-------------------------------------------------------------------------
 Insert on example  (cost=0.00..1.60 rows=6 width=92)
   ->  Foreign Scan on ossexample  (cost=0.00..1.60 rows=6 width=92)
         Foreign OssFile: osstest/example.csv.0
         Foreign OssFile Size: 728
(4 rows)
# Write the data in the example table to OSS.
insert into ossexample select * from example;
explain insert into ossexample select * from example;
                           QUERY PLAN
-----------------------------------------------------------------
 Insert on ossexample  (cost=0.00..16.60 rows=660 width=92)
   ->  Seq Scan on example  (cost=0.00..16.60 rows=660 width=92)
(2 rows)
```

Additional considerations

- oss_fdw is a foreign table plug-in developed based on the PostgreSQL FOREIGN TABLE framework.
- The data import performance varies based on the PostgreSQL cluster resources (CPU, I/O, and memory) and OSS.
- To ensure data import performance, the ApsaraDB RDS for PostgreSQL instance must be in the same region as the OSS bucket.

ID and key encryption

If the id and key parameters for CREATE SERVER are not encrypted, the `select * from pg_foreign_server` statement execution result displays the information. Your AccessKey ID and AccessKey secret are exposed. You can use symmetric encryption to hide your AccessKey ID and AccessKey secret. Use different AccessKey pairs for different instances to further protect your information. However, to avoid incompatibility with earlier versions, do not add data types as you would in Greenplum.

Encrypted information:

```
postgres=# select * from pg_foreign_server ;
  srvname  | srvowner | srvfdw | srvtype | srvversion | srvacl |
srvoptions
-----------+----------+--------+---------+------------+--------+-----------------------------------
-----------------------------------------------------------------------------------------------
---------------------------------
 ossserver |       10 |  16390 |         |            |        | {host=oss-cn-hangzhou-zmf.aliyuncs.c
om,id=MD5xxxxxxxx,key=MD5xxxxxxxx,bucket=067862}
```

The encrypted information is preceded by the MD5 hash value. The remainder of the total length divided by 8 is 3. Therefore, encryption is not performed again when the exported data is imported. You cannot create an AccessKey pair that is preceded by MD5.

12.1.16. Use Pgpool for read/write splitting in ApsaraDB RDS for PostgreSQL

This topic describes how to use the Pgpool tool of PostgreSQL installed on an ECS instance to implement read/write splitting for your primary and read-only ApsaraDB RDS for PostgreSQL instances.

Context

If you do not use Pgpool to ensure high availability, Pgpool is stateless. The decrease in performance can be ignored. Additionally, Pgpool supports horizontal scaling of your database system. You can use Pgpool and the high availability architecture of ApsaraDB RDS for PostgreSQL to implement read/write splitting.

Set up a test environment

If you have purchased a primary ApsaraDB RDS instance that runs PostgreSQL 10 and have attached read-only instances to the primary instance, you need only to install Pgpool . For more information, see Create an instance and Create a read-only ApsaraDB RDS for PostgreSQL instance. After you install Pgpool, go to Configure Pgpool.

1. Run the `vi /etc/sysctl.conf` command to open the sysctl.conf file. Modify the following configurations:

```
# add by digoal.zhou
fs.aio-max-nr = 1048576
fs.file-max = 76724600
# Optional. Set the kernel.core_pattern parameter to /data01/corefiles/core_%e_%u_%t_%s.%p.

# The /data01/corefiles directory that is used to store core dumps is created with the 777 permis
sion before testing. If a symbolic link is used, change the directory to 777.
kernel.sem = 4096 2147483647 2147483646 512000
# Specify the semaphore. You can run the ipcs -l or -u command to obtain the semaphore count. Eac
h group of 16 processes requires a semaphore with a count of 17.
kernel.shmall = 107374182
# Specify the total size of shared memory segments. Recommended value: 80% of the memory capacity
. Unit: pages.
kernel.shmmax = 274877906944
# Specify the maximum size of a single shared memory segment. Recommended value: 50% of the memor
y capacity. Unit: bytes. In PostgreSQL versions later than 9.2, the use of shared memory signific
antly drops.
kernel.shmmni = 819200
```

```
kernel.shmmni = 819200
# Specify the total number of shared memory segments that can be generated. At least two shared m
emory segments must be generated within each PostgreSQL cluster.
net.core.netdev_max_backlog = 10000
net.core.rmem_default = 262144
# The default setting of the socket receive buffer in bytes.
net.core.rmem_max = 4194304
# The maximum receive socket buffer size in bytes
net.core.wmem_default = 262144
# The default setting (in bytes) of the socket send buffer.
net.core.wmem_max = 4194304
# The maximum send socket buffer size in bytes.
net.core.somaxconn = 4096
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.tcp_keepalive_intvl = 20
net.ipv4.tcp_keepalive_probes = 3
net.ipv4.tcp_keepalive_time = 60
net.ipv4.tcp_mem = 8388608 12582912 16777216
net.ipv4.tcp_fin_timeout = 5
net.ipv4.tcp_synack_retries = 2
net.ipv4.tcp_syncookies = 1
# Enable SYN cookies. If an SYN waiting queue overflows, you can enable SYN cookies to defend aga
inst a small number of SYN attacks.
net.ipv4.tcp_timestamps = 1
# Reduce the time after which a network socket enters the TIME-WAIT state.
net.ipv4.tcp_tw_recycle = 0
# If you set this parameter to 1 to enable the recycle function, network sockets in the TIME-WAIT
state over TCP connections are recycled. However, if network address translation (NAT) is used, T
CP connections may fail. We recommend that you set this parameter to 0 on the database server.
net.ipv4.tcp_tw_reuse = 1
# Enable the reuse function. This function enables network sockets in the TIME-WAIT state to be r
eused over new TCP connections.
net.ipv4.tcp_max_tw_buckets = 262144
net.ipv4.tcp_rmem = 8192 87380 16777216
net.ipv4.tcp_wmem = 8192 65536 16777216
net.nf_conntrack_max = 1200000
net.netfilter.nf_conntrack_max = 1200000
vm.dirty_background_bytes = 409600000
# If the size of dirty pages reaches the specified limit, a background scheduling process (for ex
ample, pdflush) is invoked to flush the dirty pages to disks. These are the pages that are genera
ted n seconds earlier. The value of n is calculated by using the following formula: n = Value of
the dirty_expire_centisecs parameter/100.
# The default limit is 10% of the memory capacity. If the memory capacity is large, we recommend
that you specify the limit in bytes.
vm.dirty_expire_centisecs = 3000
# Specify the maximum period to retain dirty pages. Dirty pages are flushed to disks after the ti
me period specified by this parameter elapses. The value 3000 indicates 30 seconds.

vm.dirty_ratio = 95
# The processes that users call to write data onto disks must actively flush dirty pages to disks
. This applies when the background scheduling process to flush dirty pages is slow and the size o
f dirty pages exceeds 95% of the memory capacity. These processes include fsync and fdatasync.

# Set this parameter properly to prevent user-called processes from flushing dirty pages to disks
. This allows you to create multiple ApsaraDB RDS instances on a single server and use control gr
oups to limit the input/output operations per second (IOPS) per instance.
vm.dirty_writeback_centisecs = 100
# Specify the time interval at which the background scheduling process (such as pdflush) flushes
dirty pages to disks. The value 100 indicates 1 second.
vm.swappiness = 0
```

```
# Disable the swap function.
vm.mmap_min_addr = 65536
vm.overcommit_memory = 0
# Specify whether you can allocate more memory space than the physical host has available. If you set this parameter to 1, the system always considers the available memory space sufficient. If the memory capacity provided in the test environment is low, we recommend that you set this parameter to 1.
vm.overcommit_ratio = 90
# Specify the memory capacity that can be allocated when the overcommit_memory parameter is set to 2.
vm.swappiness = 0
# Disable the swap function.
vm.zone_reclaim_mode = 0
# Disable non-uniform memory access (NUMA). You can also disable NUMA in the vmlinux file.

net.ipv4.ip_local_port_range = 40000 65535
# Specify the range of TCP or UDP port numbers for the physical host to allocate.
fs.nr_open=20480000
# Specify the maximum number of file handles that a single process can open.
# Take note of the following parameters:
#vm.extra_free_kbytes = 4096000   # If the physical host provides a low memory capacity, do not specify a large value such as 4096000. If you specify a large value, the physical host may not start.
#vm.min_free_kbytes = 6291456    # We recommend that you increase the value of the vm.min_free_kbytes parameter by 1 GB for every 32 GB of memory.
# If the physical host does not provide much memory, we recommend that you do not configure vm.extra_free_kbytes and vm.min_free_kbytes.
# vm.nr_hugepages = 66536
# If the size of the shared buffer exceeds 64 GB, we recommend that you use huge pages. You can specify the page size by setting the Hugepagesize parameter in the /proc/meminfo file.

#vm.lowmem_reserve_ratio = 1 1 1
# If the memory capacity exceeds 64 GB, we recommend that you set this parameter. Otherwise, we recommend that you retain the default value 256 256 32.
```

2. Run the `vi /etc/security/limits.conf` command to open the limits.conf file. Modify the following configurations:

```
* soft    nofile  1024000
* hard    nofile  1024000
* soft    nproc   unlimited
* hard    nproc   unlimited
* soft    core    unlimited
* hard    core    unlimited
* soft    memlock unlimited
* hard    memlock unlimited
# Comment out the other parameters in the limits.conf file.
# Comment out the /etc/security/limits.d/20-nproc.conf file.
```

3. Run the following commands to open the rc.local file:

```
chmod +x /etc/rc.local
vi /etc/rc.local
```

Modify the following configurations to disable transparent huge pages, configure huge pages, and start PostgreSQL:

```
# Disable transparent huge pages.
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
   echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi
# Configure huge pages for two instances. Each instance has a shared buffer of 16 GB.
sysctl -w vm.nr_hugepages=17000
# Start the two instances.
su - postgres -c "pg_ctl start -D /data01/pg12_3389/pg_root"
su - postgres -c "pg_ctl start -D /data01/pg12_8002/pg_root"
```

4. Create a file system.

> **Warning** If you use a new disk, you must verify that the new disk belongs to the vdb partition instead of the vda partition. If the new disk belongs to the vda partition, data may be deleted from the new disk.

```
parted -a optimal -s /dev/vdb mklabel gpt mkpart primary 1MiB 100%FREE
mkfs.ext4 /dev/vdb1 -m 0 -O extent,uninit_bg -E lazy_itable_init=1 -b 4096 -T largefile -L vdb1
vi /etc/fstab
LABEL=vdb1 /data01 ext4 defaults,noatime,nodiratime,nodelalloc,barrier=0,data=writeback 0 0
mkdir /data01
mount -a
```

5. Start the irqbalance command line tool.

```
systemctl status irqbalance
systemctl enable irqbalance
systemctl start irqbalance
systemctl status irqbalance
```

6. Install PostgreSQL 10 and Pgpool.

```
yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
yum search all postgresql
yum search all pgpool
yum install -y postgresql12*
yum install -y pgpool-II-12-extensions
```

7. Initialize the data directory of your database system.

```
mkdir /data01/pg12_3389
chown postgres:postgres /data01/pg12_3389
```

8. Configure environment variables for the postgres user.

```
su - postgres
vi .bash_profile
```

Append the following parameters to the environment variables:

```
export PS1="$USER@`/bin/hostname -s`-> "
export PGPORT=3389
export PGDATA=/data01/pg12_$PGPORT/pg_root
export LANG=en_US.utf8
export PGHOME=/usr/pgsql-12
export LD_LIBRARY_PATH=$PGHOME/lib:/lib64:/usr/lib64:/usr/local/lib64:/lib:/usr/lib:/usr/local/li
b:$LD_LIBRARY_PATH
export DATE=`date +"%Y%m%d%H%M"`
export PATH=$PGHOME/bin:$PATH:.
export MANPATH=$PGHOME/share/man:$MANPATH
export PGHOST=$PGDATA
export PGUSER=postgres
export PGDATABASE=db1
alias rm='rm -i'
alias ll='ls -lh'
unalias vi
```

9. Initialize your primary ApsaraDB RDS instance.

```
initdb -D $PGDATA -U postgres -E UTF8 --lc-collate=C --lc-ctype=en_US.utf8
```

10. Modify the postgresql.conf file.

```
listen_addresses = '0.0.0.0'
port = 3389
max_connections = 1500
superuser_reserved_connections = 13
unix_socket_directories = '., /var/run/postgresql, /tmp'
tcp_keepalives_idle = 60
tcp_keepalives_interval = 10
tcp_keepalives_count = 10
shared_buffers = 16GB
huge_pages = on
work_mem = 8MB
maintenance_work_mem = 1GB
dynamic_shared_memory_type = posix
vacuum_cost_delay = 0
bgwriter_delay = 10ms
bgwriter_lru_maxpages = 1000
bgwriter_lru_multiplier = 10.0
bgwriter_flush_after = 512kB
effective_io_concurrency = 0
max_worker_processes = 128
max_parallel_maintenance_workers = 3
max_parallel_workers_per_gather = 4
parallel_leader_participation = off
max_parallel_workers = 8
backend_flush_after = 256
wal_level = replica
synchronous_commit = off
full_page_writes = on
wal_compression = on
wal_buffers = 16MB
wal_writer_delay = 10ms
wal_writer_flush_after = 1MB
checkpoint_timeout = 15min
max_wal_size = 64GB
min_wal_size = 8GB
checkpoint_completion_target = 0.2
checkpoint_flush_after = 256kB
```

```
random_page_cost = 1.1
effective_cache_size = 48GB
log_destination = 'csvlog'
logging_collector = on
log_directory = 'log'
log_filename = 'postgresql-%a.log'
log_truncate_on_rotation = on
log_rotation_age = 1d
log_rotation_size = 0
log_min_duration_statement = 1s
log_checkpoints = on
log_connections = on
log_disconnections = on
log_line_prefix = '%m [%p] '
log_statement = 'ddl'
log_timezone = 'Asia/Shanghai'
autovacuum = on
log_autovacuum_min_duration = 0
autovacuum_vacuum_scale_factor = 0.1
autovacuum_analyze_scale_factor = 0.05
autovacuum_freeze_max_age = 800000000
autovacuum_multixact_freeze_max_age = 900000000
autovacuum_vacuum_cost_delay = 0
vacuum_freeze_table_age = 750000000
vacuum_multixact_freeze_table_age = 750000000
datestyle = 'iso, mdy'
timezone = 'Asia/Shanghai'
lc_messages = 'en_US.utf8'
lc_monetary = 'en_US.utf8'
lc_numeric = 'en_US.utf8'
lc_time = 'en_US.utf8'
default_text_search_config = 'pg_catalog.english'
```

11. Modify the pg_hba.conf file.

> **Note** Pgpool-II is installed on the same ECS instance as the database server where PostgreSQL resides. If you specify the 127.0.0.1 IP address in the pg_hba.conf file, you must enter the correct password to ensure a successful logon.

```
# "local" is for Unix domain socket connections only
local   all             all                                     trust
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     trust
host    replication     all             127.0.0.1/32            trust
host    replication     all             ::1/128                 trust
host db123 digoal 0.0.0.0/0 md5
```

12. Execute a statement in the database to create a user authorized with streaming replication permissions. Example:

```
create role rep123 login replication encrypted password 'xxxxxxx';
```

13. Execute statements in the database to create a user and authorize it to manage your ApsaraDB RDS instances. Example:

```
create role digoal login encrypted password 'xxxxxxx';
create database db123 owner digoal;
```

14. Create a user who is authorized to check the health heartbeats between Pgpool and your read-only ApsaraDB RDS instances. With the parameters of Pgpool properly configured, this user can check the write-ahead logging (WAL) replay latency on each read-only ApsaraDB RDS instance. Example:

```
create role nobody login encrypted password 'xxxxxxx';
```

Create a secondary ApsaraDB RDS instance

To simplify the test procedure, perform the following steps to create a secondary ApsaraDB RDS instance on the same ECS instance as your primary ApsaraDB RDS instance:

1. Use the pg_basebackup tool to create a secondary ApsaraDB RDS instance.

```
pg_basebackup -D /data01/pg12_8002/pg_root -F p --checkpoint=fast -P -h 127.0.0.1 -p 3389 -U rep1
23
```

2. Run the following commands to open the postgresql.conf file of the secondary ApsaraDB RDS instance:

```
cd /data01/pg12_8002/pg_root
vi postgresql.conf
```

Modify the following configurations:

```
# The secondary ApsaraDB RDS instance has the following configurations different from the primary
ApsaraDB RDS instance:
port = 8002
primary_conninfo = 'hostaddr=127.0.0.1 port=3389 user=rep123' # You do not need to set the passwo
rd. This is because trust relationships are configured on the primary ApsaraDB RDS instance.
hot_standby = on
wal_receiver_status_interval = 1s
wal_receiver_timeout = 10s
recovery_target_timeline = 'latest'
```

3. Configure the standby.signal file of the secondary ApsaraDB RDS instance.

```
cd /data01/pg12_8002/pg_root
touch standby.signal
```

4. Execute the `SELECT * FROM pg_stat_replication ;` statement in the database to check whether data is properly synchronized between the primary and secondary ApsaraDB RDS instances. The following output is returned:

```
-[ RECORD 1 ]----+------------------------------
pid              | 21065
usesysid         | 10
usename          | postgres
application_name | walreceiver
client_addr      | 127.0.0.1
client_hostname  |
client_port      | 47064
backend_start    | 2020-02-29 00:26:28.485427+08
backend_xmin     |
state            | streaming
sent_lsn         | 0/52000060
write_lsn        | 0/52000060
flush_lsn        | 0/52000060
replay_lsn       | 0/52000060
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 0
sync_state       | async
reply_time       | 2020-02-29 01:32:40.635183+08
```

Configure Pgpool

1. Query the location where Pgpool is installed.

```
rpm -qa|grep pgpool
pgpool-II-12-extensions-4.1.1-1.rhel7.x86_64
pgpool-II-12-4.1.1-1.rhel7.x86_64
rpm -ql pgpool-II-12-4.1.1
```

2. Run the following commands to open the pgpool.conf file:

```
cd /etc/pgpool-II-12/
cp pgpool.conf.sample-stream pgpool.conf
vi pgpool.conf
```

Modify the following configurations:

```
listen_addresses = '0.0.0.0'
port = 8001
socket_dir = '/tmp'
reserved_connections = 0
pcp_listen_addresses = ''
pcp_port = 9898
pcp_socket_dir = '/tmp'
# - Backend Connection Settings -
backend_hostname0 = '127.0.0.1'
                                    # Host name or IP address to connect to for backend 0
backend_port0 = 3389
                                    # Port number for backend 0
backend_weight0 = 1
                                    # Weight for backend 0 (only in load balancing mode)
backend_data_directory0 = '/data01/pg12_3389/pg_root'
                                    # Data directory for backend 0
backend_flag0 = 'ALWAYS_MASTER'
                                    # Controls various backend behavior
                                    # ALLOW_TO_FAILOVER, DISALLOW_TO_FAILOVER
                                    # or ALWAYS_MASTER
backend_application_name0 = 'server0'
```

```
                                    # walsender's application_name, used for "show pool_nodes" com
mand
backend_hostname1 = '127.0.0.1'
backend_port1 = 8002
backend_weight1 = 1
backend_data_directory1 = '/data01/pg12_8002/pg_root'
backend_flag1 = 'DISALLOW_TO_FAILOVER'
backend_application_name1 = 'server1'
# - Authentication -
enable_pool_hba = on
                                    # Use pool_hba.conf for client authentication
pool_passwd = 'pool_passwd'
                                    # File name of pool_passwd for md5 authentication.
                                    # "" disables pool_passwd.
                                    # (change requires restart)
allow_clear_text_frontend_auth = off
                                    # Allow Pgpool-II to use clear text password authentication
                                    # with clients, when pool_passwd does not
                                    # contain the user password
# - Concurrent session and pool size -
num_init_children = 128
                                    # Number of concurrent sessions allowed
                                    # (change requires restart)
max_pool = 4
                                    # Number of connection pool caches per connection
                                    # (change requires restart)
# - Life time -
child_life_time = 300
                                    # Pool exits after being idle for this many seconds
child_max_connections = 0
                                    # Pool exits after receiving that many connections
                                    # 0 means no exit
connection_life_time = 0
                                    # Connection to backend closes after being idle for this many
seconds
                                    # 0 means no close
client_idle_limit = 0
                                    # Client is disconnected after being idle for that many second
s
                                    # (even inside an explicit transactions!)
                                    # 0 means no disconnection
#------------------------------------------------------------------------------
# LOGS
#------------------------------------------------------------------------------
# - Where to log -
log_destination = 'syslog'
                                    # Where to log
                                    # Valid values are combinations of stderr,
                                    # and syslog. Default to stderr.
log_connections = on
                                    # Log connections
log_standby_delay = 'if_over_threshold'
                                    # Log standby delay
                                    # Valid values are combinations of always,
                                    # if_over_threshold, none
#------------------------------------------------------------------------------
# FILE LOCATIONS
#------------------------------------------------------------------------------
pid_file_name = '/var/run/pgpool-II-12/pgpool.pid'
                                    # PID file name
```

```
                                    # PID file name
                                    # Can be specified as relative to the"
                                    # location of pgpool.conf file or
                                    # as an absolute path
                                    # (change requires restart)
logdir = '/tmp'
                                    # Directory of pgPool status file
                                    # (change requires restart)
#------------------------------------------------------------------------------
# CONNECTION POOLING
#------------------------------------------------------------------------------
connection_cache = on
                                    # Activate connection pools
                                    # (change requires restart)
                                    # Semicolon separated list of queries
                                    # to be issued at the end of a session
                                    # The default is for 8.3 and later
reset_query_list = 'ABORT; DISCARD ALL'
#------------------------------------------------------------------------------
# LOAD BALANCING MODE
#------------------------------------------------------------------------------
load_balance_mode = on
                                    # Activate load balancing mode
                                    # (change requires restart)
ignore_leading_white_space = on
                                    # Ignore leading white spaces of each query
white_function_list = ''
                                    # Comma separated list of function names
                                    # that don't write to database
                                    # Regexp are accepted
black_function_list = 'currval,lastval,nextval,setval'
                                    # Comma separated list of function names
                                    # that write to database
                                    # Regexp are accepted
black_query_pattern_list = ''
                                    # Semicolon separated list of query patterns
                                    # that should be sent to primary node
                                    # Regexp are accepted
                                    # valid for streaming replicaton mode only.
database_redirect_preference_list = ''
                                    # comma separated list of pairs of database and node id.
                                    # example: postgres:primary,mydb[0-4]:1,mydb[5-9]:2'
                                    # valid for streaming replicaton mode only.
app_name_redirect_preference_list = ''
                                    # comma separated list of pairs of app name and node id.
                                    # example: 'psql:primary,myapp[0-4]:1,myapp[5-9]:standby'
                                    # valid for streaming replicaton mode only.
allow_sql_comments = off
                                    # if on, ignore SQL comments when judging if load balance or
                                    # query cache is possible.
                                    # If off, SQL comments effectively prevent the judgment
                                    # (pre 3.4 behavior).
disable_load_balance_on_write = 'transaction'
                                    # Load balance behavior when write query is issued
                                    # in an explicit transaction.
                                    # Note that any query not in an explicit transaction
                                    # is not affected by the parameter.
                                    # 'transaction' (the default): if a write query is issued,
                                    # subsequent read queries will not be load balanced
                                    # until the transaction ends.
```

```
                                    # 'trans_transaction': if a write query is issued,
                                    # subsequent read queries in an explicit transaction
                                    # will not be load balanced until the session ends.
                                    # 'always': if a write query is issued, read queries will
                                    # not be load balanced until the session ends.
statement_level_load_balance = off
                                    # Enables statement level load balancing
#------------------------------------------------------------------------------
# MASTER/SLAVE MODE
#------------------------------------------------------------------------------
master_slave_mode = on
                                    # Activate master/slave mode
                                    # (change requires restart)
master_slave_sub_mode = 'stream'
                                    # Master/slave sub mode
                                    # Valid values are combinations stream, slony
                                    # or logical. Default is stream.
                                    # (change requires restart)
# - Streaming -
sr_check_period = 3
                                    # Streaming replication check period
                                    # Disabled (0) by default
sr_check_user = 'nobody'
                                    # Streaming replication check user
                                    # This is neccessary even if you disable streaming
                                    # replication delay check by sr_check_period = 0
sr_check_password = ''
                                    # Password for streaming replication check user
                                    # Leaving it empty will make Pgpool-II to first look for the
                                    # Password in pool_passwd file before using the empty password

sr_check_database = 'postgres'
                                    # Database name for streaming replication check
delay_threshold = 512000
                                    # Threshold before not dispatching query to standby node
                                    # Unit is in bytes
                                    # Disabled (0) by default
#------------------------------------------------------------------------------
# HEALTH CHECK GLOBAL PARAMETERS
#------------------------------------------------------------------------------
health_check_period = 5
                                    # Health check period
                                    # Disabled (0) by default
health_check_timeout = 10
                                    # Health check timeout
                                    # 0 means no timeout
health_check_user = 'nobody'
                                    # Health check user
health_check_password = ''
                                    # Password for health check user
                                    # Leaving it empty will make Pgpool-II to first look for the
                                    # Password in pool_passwd file before using the empty password

health_check_database = ''
                                    # Database name for health check. If '', tries 'postgres' fris
t,
health_check_max_retries = 60
                                    # Maximum number of times to retry a failed health check befor
e giving up.
```

```
health_check_retry_delay = 1
                                    # Amount of time to wait (in seconds) between retries.
connect_timeout = 10000
                                    # Timeout value in milliseconds before giving up to connect to
backend.
                                    # Default is 10000 ms (10 second). Flaky network user may want
to increase
                                    # the value. 0 means no timeout.
                                    # Note that this value is not only used for health check,
                                    # but also for ordinary conection to backend.
#------------------------------------------------------------------------------
# FAILOVER AND FAILBACK
#------------------------------------------------------------------------------
failover_on_backend_error = off
                                    # Initiates failover when reading/writing to the
                                    # backend communication socket fails
                                    # If set to off, pgpool will report an
                                    # error and disconnect the session.
relcache_expire = 0  # After the configuration file is restructured, we recommend that you set th
is parameter to 1, reload the configuration file, and then set this parameter to 0 again. You can
also set this parameter to a specific point in time.
                                    # Life time of relation cache in seconds.
                                    # 0 means no cache expiration(the default).
                                    # The relation cache is used for cache the
                                    # query result against PostgreSQL system
                                    # catalog to obtain various information
                                    # including table structures or if it's a
                                    # temporary table or not. The cache is
                                    # maintained in a pgpool child local memory
                                    # and being kept as long as it survives.
                                    # If someone modify the table by using
                                    # ALTER TABLE or some such, the relcache is
                                    # not consistent anymore.
                                    # For this purpose, cache_expiration
                                    # controls the life time of the cache.
relcache_size = 8192
                                    # Number of relation cache
                                    # entry. If you see frequently:
                                    # "pool_search_relcache: cache replacement happend"
                                    # in the pgpool log, you might want to increate this number.
```

3. Run the `cd /etc/pgpool-II-12` command to configure the pool_passwd file.

> **Note** If you connect to your ApsaraDB RDS instances by using Pgpool, you must configure the pool_passwd file. This is because Pgpool supports the authentication protocol of PostgreSQL.

```
# Run the following command:
#pg_md5 --md5auth --username=username password
# Generate the passwords of the digoal and nobody users. The passwords are automatically written
into the pool_passwd file.
pg_md5 --md5auth --username=digoal "xxxxxxx"
pg_md5 --md5auth --username=nobody "xxxxxxx"
```

4. Use the system to automatically generate the pool_passwd file.

```
cd /etc/pgpool-II-12
cat pool_passwd
```

5. Run the following commands to configure the pgpool_hba file:

```
cd /etc/pgpool-II-12
cp pool_hba.conf.sample pool_hba.conf
vi pool_hba.conf
```

Configure the following parameters:

```
host all all 0.0.0.0/0 md5
```

6. Configure the pcp.conf file.

> **Note** The pcp.conf file is used to manage the users and passwords of Pgpool. It is not related to the users and passwords of your ApsaraDB RDS instances.

```
cd /etc/pgpool-II-12
# pg_md5 abc  # In this command, you set the password to abc and encrypt it by using the MD5 encr
yption algorithm.
900150983cd24fb0d6963f7d28e17f72
cp pcp.conf.sample pcp.conf
vi pcp.conf
# USERID:MD5PASSWD
manage:900150983cd24fb0d6963f7d28e17f72  # In this command, the manage user is used to manage PCP
.
```

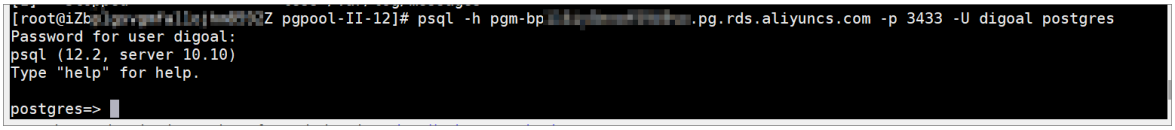
7. Start Pgpool.

```
cd /etc/pgpool-II-12
pgpool -f ./pgpool.conf -a ./pool_hba.conf -F ./pcp.conf
```

> **Note** If you want to view the logs of Pgpool, run the following command:
>
> ```
> less /var/log/messages
> ```

8. Use Pgpool to connect to your ApsaraDB RDS instances.

```
psql -h 127.0.0.1 -p 8001 -U digoal postgres
```



FAQ

- Q: How do I test whether read/write splitting is enabled?

  A: You can connect to your ApsaraDB RDS instances by using Pgpool and call the pg_is_in_recovery() function. Then, close the connection, establish a connection again, and call the pg_is_in_recovery() function again. If you receive a value of false and then a value of true, Pgpool routes requests to your primary ApsaraDB RDS instance and then to your read-only ApsaraDB RDS instances, and read/write splitting is enabled.

- Q: Does Pgpool increase the latency?

  A: Pgpool increases the latency slightly. In the test environment you set up in this topic, the latency increases by about 0.12 milliseconds.

- Q: How does Pgpool check the latency and health on my read-only ApsaraDB RDS instances?

- A: If the WAL replay latency on a read-only ApsaraDB RDS instance exceeds the specified limit, Pgpool stops routing SQL requests to the read-only instance. Pgpool resumes routing SQL requests to the read-only instance only after it detects that the WAL replay latency on the read-only instance falls below the specified limit.

  > **Note** Connect to your primary ApsaraDB RDS instance and query the location where the current WAL data record is written. This location is referred to as log sequence number (LSN) 1. Then, connect to a read-only ApsaraDB RDS instance and query the location where the current WAL data record is replayed. This location is referred to as LSN 2. You can obtain the number of bytes between LSN 1 and LSN 2. This number indicates the latency.

- Pgpool monitors the health of your read-only ApsaraDB RDS instances. If a read-only instance is unhealthy, Pgpool stops routing requests to the read-only instance.

- Q: How do I stop Pgpool and reload the configuration of Pgpool?

  A: Run the `pgpool --help` command to obtain more information about the commands used in Pgpool. Example:

  ```
  cd /etc/pgpool-II-12
  pgpool -f ./pgpool.conf -m fast stop
  ```

- Q: How do I configure Pgpool if more than one read-only ApsaraDB RDS instance is attached to my primary ApsaraDB RDS instance?

  A: Add the configurations of all the attached read-only ApsaraDB RDS instances to the pgpool.conf file. Example:

  ```
  backend_hostname1 = 'xx.xx.xxx.xx'
  backend_port1 = 8002
  backend_weight1 = 1
  backend_data_directory1 = '/data01/pg12_8002/pg_root'
  backend_flag1 = 'DISALLOW_TO_FAILOVER'
  backend_application_name1 = 'server1'
  backend_hostname2 = 'xx.xx.xx.xx'
  backend_port1 = 8002
  backend_weight1 = 1
  backend_data_directory1 = '/data01/pg12_8002/pg_root'
  backend_flag1 = 'DISALLOW_TO_FAILOVER'
  backend_application_name1 = 'server1'
  ```

- Q: How do I use pcp commands to view the status of my read-only ApsaraDB RDS instances?

  A: To obtain the status of your read-only ApsaraDB RDS instances by using pcp commands, run the following command:

  ```
  # pcp_node_info -U manage -h /tmp -p 9898 -n 1 -v
  Password: # Enter the password.
  Hostname               : 127.0.0.1
  Port                   : 8002
  Status                 : 2
  Weight                 : 0.500000
  Status Name            : up
  Role                   : standby
  Replication Delay      : 0
  Replication State      :
  Replication Sync State :
  Last Status Change     : 2020-02-29 00:20:29
  ```

- Q: Which listening ports are used by Pgpool for read/write splitting?

A: The following listening ports are used by Pgpool for read/write splitting:

- Primary ApsaraDB RDS instance: Port 3389
- Secondary ApsaraDB RDS instance: Port 8002
- Pgpool: Port 8001
- PCP: Port 9898

12.1.17. Use ShardingSphere to develop ApsaraDB RDS for PostgreSQL

ShardingSphere is an open source ecosystem that consists of a set of distributed database middleware solutions.

Prerequisites

All PostgreSQL versions used with ApsaraDB RDS support ShardingSphere.

Context

ApsaraDB RDS for PostgreSQL supports database-integrated sharding plug-ins (such as Citus, Postgres-XC, and AntDB) and massively parallel processing (MPP) products. It also supports sharding middleware products that are similar to those widely used in MySQL, such as ShardingSphere.

ShardingSphere is suitable for services that run in databases with thorough, well-organized logical sharding. It offers the following features:

- Data sharding
  - Database and table sharding
  - Read/write splitting
  - Sharding strategy customization
  - Decentralized distributed primary key
- Distributed transaction
  - Unified transaction API
  - XA transaction
  - BASE transaction
- Database orchestration
  - Dynamic configuration
  - Orchestration and governance
  - Data encryption
  - Tracing and observability
  - Elastic scaling out (planning)

For more information, visit the ShardingSphere documentation.

ShardingSphere products

ShardingSphere includes three independent products. You can choose the product that best suits your business requirements. The following table describes these products.

Parameter	Sharding-JDBC	Sharding-Proxy	Sharding-Sidecar

Parameter	Sharding-JDBC	Sharding-Proxy	Sharding-Sidecar
Supported database engine	All JDBC-compatible database engines such as MySQL, PostgreSQL, Oracle, and SQL Server	MySQL and PostgreSQL	MySQL and PostgreSQL
Connections consumed	High	Low	High
Supported heterogeneous language	Java	All	All
Performance	Low consumption	Moderate consumption	Low consumption
Decentralized	Yes	No	Yes
Stateless API	No	Yes	No

Prepare configuration templates

1. On your ECS instance, run the following command to go to the directory where configuration templates are stored. The directory is under the root directory in this example.

```
cd /root/apache-shardingsphere-incubating-4.0.0-sharding-proxy-bin/conf
```

2. Run the `ll` command to view all files stored in the directory: Command output:

```
total 24
-rw-r--r-- 1 501 games 3019 Jul 30  2019 config-encrypt.yaml
-rw-r--r-- 1 501 games 3582 Apr 22  2019 config-master_slave.yaml
-rw-r--r-- 1 501 games 4278 Apr 22  2019 config-sharding.yaml
-rw-r--r-- 1 501 games 1918 Jul 30  2019 server.yaml
```

> **Note**
> - config-encrypt.yaml: the data encryption configuration file.
> - config-master_slave.yaml: the read/write splitting configuration file.
> - config-sharding.yaml: the data sharding configuration file.
> - server.yaml: the common configuration file.

3. Modify the configuration files.

> **Note** For more information about the configuration files, visit the ShardingSphere documentation. In this example, the data sharding and common configuration files are used.
> - Example of a data sharding configuration file:

```
schemaName: # The name of the logical data source.
dataSources: # The configuration of the data source. You can configure more than one data source by using the data_source_name variable.
  <data_source_name>: # You do not need to configure a database connection pool. This is different in Sharding-JDBC.
    url: # The URL used to connect to your database.
    username: # The username used to log on to the database.
    password: # The password used to log on to the database.
    connectionTimeoutMilliseconds: 30000 # The connection timeout period in milliseconds.
    idleTimeoutMilliseconds: 60000 # The idle-connection reclaiming timeout period in milliseconds.
    maxLifetimeMilliseconds: 1800000 # The maximum connection time to live (TTL) in milliseconds.
    maxPoolSize: 65 # The maximum number of connections allowed.
shardingRule: # You do not need to configure a sharding rule, because it is the same in Sharding-JDBC.
```

- Example of a common configuration file:

```
Proxy properties
# You do not need to configure proxy properties that are the same in Sharding-JDBC
props:
  acceptor.size: # The number of worker threads that receive requests from the client. The default number is equal to the number of CPU cores multiplied by 2.
  proxy.transaction.type: # The type of transaction processed by the proxy. Valid values: LOCAL | XA | BASE. Default value: LOCAL. Value XA specifies to use Atomikos as the transaction manager. Value BASE specifies to copy the .jar package that implements the ShardingTransactionManager operation to the lib directory.
  proxy.opentracing.enabled: # Specifies whether to enable link tracing. Link tracing is disabled by default.
  check.table.metadata.enabled: # Specifies whether to check the consistency of metadata among sharding tables during startup. Default value: false.
  proxy.frontend.flush.threshold: # The number of packets returned in a batch during a complex query.
Permission verification
This part of the configuration is used to verify your permissions when you attempt to log on to Sharding-Proxy. After you configure the username, password, and authorized databases, you must use the correct username and password to log on to Sharding-Proxy from the authorized databases.
authentication:
  users:
    root: # The username of the root user.
      password: root# The password of the root user.
    sharding: # The username of the sharding user.
      password: sharding# The password of the sharding user.
      authorizedSchemas: sharding_db, masterslave_db # The databases in which the specified user is authorized. If you want to specify more than one database, separate them with commas (,). You are granted the permissions of the root user by default. This way, you can access all databases.
```

Set up a test environment

- On your ECS instance, install Java.

```
yum install -y java
```

- Configure an ApsaraDB RDS instance that runs PostgreSQL 10.

- Create an account with username r1.
- Set the password of the account to "PW123321!".
- Create the following databases whose owners are user r1: db0, db1, db2, and db3.
- Add the IP address of your ECS instance to an IP address whitelist of the ApsaraDB RDS for PostgreSQL instance.

> **Note**
> - For more information about how to create an ApsaraDB RDS for PostgreSQL instance, database, and account, see Create an instance and Create a database and an account.
> - For more information about how to configure an IP address whitelist, see Configure an IP address whitelist.

- Run `vi /root/apache-shardingsphere-incubating-4.0.0-sharding-proxy-bin/conf/server.yaml` to configure the following common configuration file:

```
  authentication:
  users:
    r1:
      password: PW123321!
      authorizedSchemas: db0,db1,db2,db3
props:
  executor.size: 16
  sql.show: false
```

Test horizontal sharding

1. Run `vi /root/apache-shardingsphere-incubating-4.0.0-sharding-proxy-bin/conf/config-sharding.yaml` to modify the following data sharding configuration file:

```
schemaName: sdb
dataSources:
  db0:
    url: jdbc:postgresql://pgm-bpxxxxx.pg.rds.aliyuncs.com:1433/db0
    username: r1
    password: PW123321!
    connectionTimeoutMilliseconds: 30000
    idleTimeoutMilliseconds: 60000
    maxLifetimeMilliseconds: 1800000
    maxPoolSize: 65
  db1:
    url: jdbc:postgresql://pgm-bpxxxxx.pg.rds.aliyuncs.com:1433/db1
    username: r1
    password: PW123321!
    connectionTimeoutMilliseconds: 30000
    idleTimeoutMilliseconds: 60000
    maxLifetimeMilliseconds: 1800000
    maxPoolSize: 65
  db2:
    url: jdbc:postgresql://pgm-bpxxxxx.pg.rds.aliyuncs.com:1433/db2
    username: r1
    password: PW123321!
    connectionTimeoutMilliseconds: 30000
    idleTimeoutMilliseconds: 60000
    maxLifetimeMilliseconds: 1800000
    maxPoolSize: 65
  db3:
    url: jdbc:postgresql://pgm-bpxxxxx.pg.rds.aliyuncs.com:1433/db3
```

```
    username: r1
    password: PW123321!
    connectionTimeoutMilliseconds: 30000
    idleTimeoutMilliseconds: 60000
    maxLifetimeMilliseconds: 1800000
    maxPoolSize: 65
shardingRule:
  tables:
    t_order:
      actualDataNodes: db${0..3}.t_order${0..7}
      databaseStrategy:
        inline:
          shardingColumn: user_id
          algorithmExpression: db${user_id % 4}
      tableStrategy:
        inline:
          shardingColumn: order_id
          algorithmExpression: t_order${order_id % 8}
      keyGenerator:
        type: SNOWFLAKE
        column: order_id
    t_order_item:
      actualDataNodes: db${0..3}.t_order_item${0..7}
      databaseStrategy:
        inline:
          shardingColumn: user_id
          algorithmExpression: db${user_id % 4}
      tableStrategy:
        inline:
          shardingColumn: order_id
          algorithmExpression: t_order_item${order_id % 8}
      keyGenerator:
        type: SNOWFLAKE
        column: order_item_id
  bindingTables:
    - t_order,t_order_item
  defaultTableStrategy:
    none:
```

2. Start ShardingSphere and listen to Port 8001.

```
cd /root/apache-shardingsphere-incubating-4.0.0-sharding-proxy-bin/bin/
./start.sh 8001
```

3. Connect to the destination database.

```
psql -h 127.0.0.1 -p 8001 -U r1 sdb
```

4. Create a table.

```
create table t_order(order_id int8 primary key, user_id int8, info text, c1 int, crt_time timesta
mp);
create table t_order_item(order_item_id int8 primary key, order_id int8, user_id int8, info text,
c1 int, c2 int, c3 int, c4 int, c5 int, crt_time timestamp);
```

> **Note** When you create a table, the system creates horizontal shards in the destination database based on the sharding strategy that you specify.

FAQ

- If you want to view the SQL parsing and routing statements used in ShardingSphere, run `vi /root/apache-shardingsphere-incubating-4.0.0-sharding-proxy-bin/conf/server.yaml`.

```
  authentication:
  users:
    r1:
      password: PW123321!
      authorizedSchemas: db0,db1,db2,db3
props:
  executor.size: 16
  sql.show: true # Specifies to log parsed SQL statements.
```

- If you want to test writes and queries, run the following commands:

```
insert into t_order (user_id, info, c1, crt_time) values (0,'a',1,now());
insert into t_order (user_id, info, c1, crt_time) values (1,'b',2,now());
insert into t_order (user_id, info, c1, crt_time) values (2,'c',3,now());
insert into t_order (user_id, info, c1, crt_time) values (3,'c',4,now());
select * from t_order;
```

  The following result is returned in this example:

```
      order_id      | user_id | info | c1 |          crt_time
--------------------+---------+------+----+----------------------------
 433352561047633921 |       0 | a    |  1 | 2020-02-09 19:48:21.856555
 433352585668198400 |       1 | b    |  2 | 2020-02-09 19:48:27.726815
 433352610813050881 |       2 | c    |  3 | 2020-02-09 19:48:33.721754
 433352628370407424 |       3 | c    |  4 | 2020-02-09 19:48:37.907683
(4 rows)
```

- If you want to view ShardingSphere logs, run the following command:

```
/root/apache-shardingsphere-incubating-4.0.0-sharding-proxy-bin/logs/stdout.log
```

- If you want to use pgbench for stress testing, run the following commands:

```
vi test.sql
\set user_id random(1,100000000)
\set order_id random(1,2000000000)
\set order_item_id random(1,2000000000)
insert into t_order (user_id, order_id, info, c1 , crt_time) values (:user_id, :order_id,random():
:text, random()*1000, now()) on conflict (order_id) do update set info=excluded.info,c1=excluded.c
1,crt_time=excluded.crt_time;
insert into t_order_item (order_item_id, user_id, order_id, info, c1,c2,c3,c4,c5,crt_time) values
(:order_item_id, :user_id,:order_id,random()::text, random()*1000,random()*1000,random()*1000,rand
om()*1000,random()*1000, now()) on conflict(order_item_id) do nothing;
pgbench -M simple -n -r -P 1 -f ./test.sql -c 24 -j 24  -h 127.0.0.1 -p 8001 -U r1 sdb -T 120
progress: 1.0 s, 1100.9 tps, lat 21.266 ms stddev 6.349
progress: 2.0 s, 1253.0 tps, lat 18.779 ms stddev 7.913
progress: 3.0 s, 1219.0 tps, lat 20.083 ms stddev 13.212
```

13.Cloud Native Distributed Database PolarDB-X

13.1. User Guide (1.0)

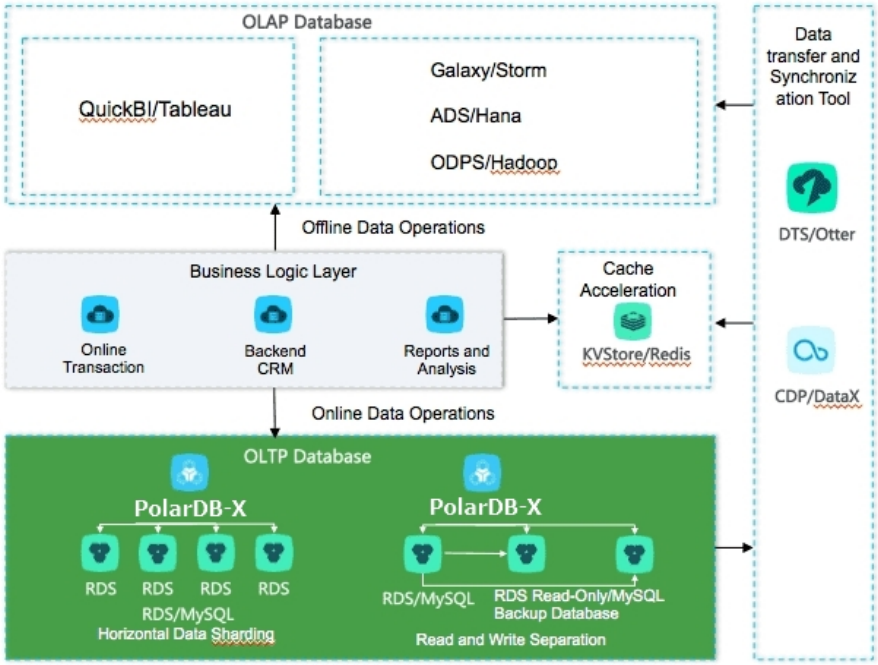
13.1.1. What is PolarDB-X?

is a database product that is developed by Alibaba Group and focuses on scaling out single-instance relational databases. This service is compatible with former Distributed Relational Database Service (DRDS).

Compatible with the MySQL communication protocols, supports most MySQL data manipulation language (DML) and data definition language (DDL) syntax. It provides the core capabilities and features of distributed databases, such as sharding, smooth scale-out, service upgrade and downgrade, and transparent read/write splitting. PolarDB-X is lightweight (stateless), flexible, stable, and efficient, and provides you with O&M capabilities throughout the lifecycle of distributed databases.

is used for operations on large-scale online data. PolarDB-X maximizes the operation efficiency by partitioning data in specific business scenarios. This meets the requirements of online business on relational databases in an effective way.

Figure of the architecture



Fixed issues

- Capacity bottlenecks of single-instance databases: As the data volume and access increase, traditional single-instance databases encounter great challenges that cannot be solved by hardware upgrades in a complete way. In distributed database solutions, multiple instances work in a joint way. This resolves the bottlenecks of data storage capacity and access volumes in an effective way.
- Difficult scale-out of relational databases: Due to the inherent attributes of distributed databases, you can change the shards where data is stored through smooth data migration. This way, the dynamic scale-out of relational databases is achieved.

13.1.2. Quick start

This topic describes how to get started with .

A instance is physically a distributed cluster that consists of multiple server nodes and underlying storage instances. A database is a logical concept and only contains metadata. Specific data is stored in the physical database of the underlying storage instance. To get started with , follow these steps:

1. Create a PolarDB-X instance.
2. Create a database.

   To create a database in a instance, you must select one or more ApsaraDB RDS for MySQL instances as the data storage nodes. If no RDS instance exists, create one first. For more information about how to create and manage ApsaraDB RDS for MySQL instances, see *User Guide of RDS* .
3. After a database is created, you also need to create tables in the database like in a single-instance database. However, the syntax is different, mainly in the expression of data partitioning information in the table creation statement. For more information about how to create a table, see Table creation syntax.

13.1.3. Log on to the PolarDB-X console

This topic describes how to log on to the console by using Google Chrome.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %
3. Click **Log On**.
4. In the top navigation bar, choose **Products > Database Services > Distributed Relational Database Service**.

13.1.4. Instance management

13.1.4.1. Create a PolarDB-X instance

To use , you must first create an instance. This topic describes how to create a instance.

1. Log on to the PolarDB-X console.
2. On the page that appears, click **Create Instance** in the upper-right corner.

3. On the **Create DRDS Instance** page, set parameters as required.

   Parameters for creating a PolarDB-X instance describes the parameters.

   Parameters for creating a PolarDB-X instance

Type	Parameter	Description
Region	**Organization**	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
	Region	The region where the instance resides. Services in different regions are not interconnected over the internal network. After the instance is created, the region cannot be changed.
	Zone	The zone where the instance resides.
Basic Settings	**Instance Type**	The type of the instance. Select an instance type from the options available on the page.
	Instance Edition	The edition of the instance. Valid values: ○ Standard ○ Enterprise ○ Starter
	Instance Specifications	The specifications of the instance. The rules vary with instance editions. Select the instance specifications from the options available on the page.
Network Type	**Network Type**	The network type of the instance. instances support the following network types: ○ **Classic Network**: Cloud services on a classic network are not isolated from each other. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ○ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can customize routing tables, IP address ranges, and gateways in a VPC. We recommend that you select VPC for higher security. Select VPC for Network Type, and then set **VPC** and **VSwitch**. **Note** Make sure that the instance has the same network type as the Elastic Compute Service (ECS) instance to which you want to connect. If the PolarDB-X and ECS instances have different network types, they cannot communicate over an internal network.

4. Click **Submit**.

   After the instance is created, it appears in the instance list and its status changes to **Running**. An instance name uniquely identifies a instance.

13.1.4.2. Change specifications

When you use , you can change the specifications of a instance as needed.

Procedure

1. Log on to the PolarDB-X console.
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. On the Basic Information page, click **Upgrade** or **Downgrade** in the **Common Operations** section to access the Change Specifications page.

   > **Note** Alternatively, on the **DRDS Instance Management** page, choose **More > Downgrade** from the **Actions** column of the target instance.

5. On the Change Specifications page, set Instance Edition and Instance Specifications, and then click **Submit**.

   After a few minutes, you can view the new specifications of the instance in the instance list.

   > **Note** Specifications downgrade leads to transient disconnections between applications and within a short period of time. Make sure that your applications can be automatically reconnected.

13.1.4.3. Read-only PolarDB-X instances

13.1.4.3.1. Overview

Read-only instances are extension and supplement to primary instances and are compatible with SQL query syntax of primary instances.

Features

Read-only and primary instances can share the same replica of data. You can perform complex data query and analysis directly on read-only or primary ApsaraDB RDS for MySQL instances. Multiple instance types are provided to handle highly concurrent access requests and reduce the response time (RT) for complex queries. Resource isolation alleviates the load pressure on the primary instances and reduces the link complexity of the business architecture. It reduces the O&M and budget costs, eliminating the need for additional data synchronization.

Instance type

Concurrent read-only instances: For high-concurrency and high-traffic simple queries or offline data extraction, resource isolation protects you against highly concurrent queries, ensuring the stability of online business links.

> **Note** For the businesses with primary instances, concurrent read-only instances can be used in the following scenarios:
>
> - High-concurrency and high-traffic simple queries are performed.
> - Data is extracted offline.

Limits

- Primary and read-only instances must be in the same region, but they can be in different zones.
- A read-only instance must belong to a primary instance. Before creating a read-only instance, you must create a primary instance. After you create a database on the primary instance, the database is replicated to the read-only instance. If you delete the database from the primary instance, the corresponding database on the read-only instance is also deleted.
- You are not allowed to migrate data to read-only instances.
- You are not allowed to create or delete databases in read-only instances.

- read-only instances cannot be cloned.
- read-only instances support data definition language (DDL) statements but do not support data manipulation language (DML) statements for data modification.

13.1.4.3.2. Create a read-only PolarDB-X instance

This topic describes how to create a read-only Cloud Native Distributed Database PolarDB-X (PolarDB-X) instance.

Procedure

1. Log on to the PolarDB-X console.
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. On the Basic Information page, click **Create DRDS Read-only Instance** in the Related Instances section.
5. Set Region, Basic Settings, and Network Type, and then click **Submit**.

   Parameters for creating a read-only PolarDB-X instance

Type	Parameter	Description
Region	**Region**	The region where the read-only instance resides. Services in different regions are not interconnected over the internal network. After the instance is created, the region cannot be changed.
	Zone	The zone where the read-only instance resides.
Basic Settings	**Instance Type**	The type of the read-only instance. Select an instance type from the options available on the page.
	Instance Edition	The edition of the read-only instance. Valid values: ○ Starter ○ Standard ○ Enterprise
	Instance Specifications	The specifications of the read-only instance. The rules vary with instance editions. Select the instance specifications from the options available on the page.
	Description	The description of the read-only instance. We recommend that you provide an informative description to simplify future management operations.

Type	Parameter	Description
Network Type	**Network Type**	The network type of the read-only instance. instances support the following network types: ○ **Classic Network**: Cloud services on a classic network are not isolated from each other. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ○ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can customize routing tables, IP address ranges, and gateways in a VPC. We recommend that you select VPC for higher security. Select VPC for Network Type, and then set **VPC** and **VSwitch**. **Note** Make sure that the instance has the same network type as the Elastic Compute Service (ECS) instance to which you want to connect. If the PolarDB-X and ECS instances have different network types, they cannot communicate over an internal network.

6. It takes several minutes to create the instance. Please wait. After the instance is created, it appears in the instance list in the console.

13.1.4.3.3. Manage a read-only PolarDB-X instance

Read-only instances are managed in a similar way as primary instances. However, databases cannot be created or deleted on the read-only instance management page. Databases on read-only instances are created or deleted with databases on primary instances. In the console, you can go to the read-only instance management page in two ways.

Manage a read-only PolarDB-X instance by its ID

1. Log on to the PolarDB-X console.
2. On the **DRDS Instance Management** page, find the target read-only instance.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the target read-only instance to access the **Basic Information** page.

Manage a read-only PolarDB-X instance by the ID of its primary instance

1.
2. On the **DRDS Instance Management** page, find the target primary instance.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the target primary instance to access the **Basic Information** page.
4. On the **Basic Information** page, move the pointer over the number of read-only instances in the **Related Instances** section to view the ID of the read-only PolarDB-X instance.
5. Click the ID of the target read-only PolarDB-X instance. The **Basic Information** page of the read-only instance appears.

13.1.4.3.4. Release a read-only PolarDB-X instance

If you no longer need a read-only instance, you can release it.

Prerequisites

The read-only instance must be in the **Running** state.

Procedure

1. Log on to the PolarDB-X console.
2. Find the target instance in the instance list.
3. In the PolarDB-X instance list, find the target instance, and choose **More > Release** from the Actions column.

   > **Notice** You cannot recover the PolarDB-X instances that have been released. Exercise caution when you perform this operation.

4. In the **Release DRDS Instance** dialog box, click **OK**.

13.1.4.4. Restart a PolarDB-X instance

This topic describes how to restart a instance.

Prerequisites

The PolarDB-X instance must be in the **Running** state.

Procedure

1. Log on to the PolarDB-X console.
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. Click **Restart Instance** in the upper-right corner.
5. In the **Restart Instance** dialog box, click **OK**.

   > **Notice** Restarting a PolarDB-X instance terminates all its connections. Make appropriate service arrangements before you restart a PolarDB-X instance. Exercise caution when you perform this operation.

13.1.4.5. Release a PolarDB-X instance

This topic describes how to release a running PolarDB-X instance in the console.

Prerequisites

- All databases on the PolarDB-X instance have been deleted.
- The PolarDB-X instance must be in the Running state.

Procedure

1. Log on to the PolarDB-X console.
2. Find the target instance in the instance list.
3. In the PolarDB-X instance list, find the target instance, and choose **More > Release** from the Actions column.
4. In the **Release DRDS Instance** dialog box, click **OK**.

> **Warning** After the PolarDB-X instance is released, data is not deleted from its attached ApsaraDB RDS for MySQL instances. However, a released PolarDB-X instance cannot be restored. Exercise caution when you perform this operation.

13.1.4.6. Recover data

13.1.4.6.1. Backup and restoration

allows you to back up data of instances and databases and restore them by using the backup data. Instances can be automatically or manually backed up. PolarDB-X provides fast backup and consistent backup. Existing backup sets are used to restore data in instances. Data is restored to the new and ApsaraDB RDS for MySQL instances by using the existing backup sets.

Considerations

- By default, the automatic backup policy of is disabled. You must manually enable it.
- The log backup capability of relies on underlying ApsaraDB RDS for MySQL instances. Therefore, the log backup policy configured in the console is automatically synchronized to all underlying ApsaraDB RDS for MySQL instances. After the policy is configured, do not modify it in the ApsaraDB RDS console. Otherwise, related data backup sets may be invalid.
- The backup and restoration feature of relies on log backup. We recommend that you enable the log backup policy by default to prevent backup sets from becoming invalid.
- Data definition language (DDL) operations cannot be performed during the backup process. Otherwise, instance backup and restoration may fail.
- During data backup, ensure that the underlying ApsaraDB RDS for MySQL instances for the instance are normal. Otherwise, data backup may fail.
- Consistent backup and restoration is supported only by 5.3.8 and later.
- Ensure that all tables have primary keys. Otherwise, data accuracy may be affected during consistent backup and restoration.
- During consistent backup, distributed transactions on instances are locked for seconds. During the locking period, the execution of non-transactional SQL statements and non-distributed transactions is not affected. However, the commitment of distributed transactions is blocked and the response time (RT) for executing SQL statements may have millisecond-level jitters. We recommend that you perform consistent backup during off-peak hours.
- Due to changes in the inventory of and ApsaraDB RDS for MySQL resources, automatically adjusts the instance type and zone during instance restoration. We recommend that you confirm and adjust the instance type and zone after the instance restoration to avoid business disruption.

Backup methods

For different scenarios, provides fast backup and consistent backup and the related data restoration capabilities. The following table compares the two backup methods.

Backup method	Scenario	Benefit	Disadvantage
Fast backup	Applies to routine backup and restoration scenarios.	• Fast data backup and restoration is enabled. • Data can be restored by backup set or by time. • All instance versions support this feature.	In sharding scenarios, data consistency can be ensured only within a single ApsaraDB RDS for MySQL instance. Global data consistency cannot be ensured.

Backup method	Scenario	Benefit	Disadvantage
Consistent backup	Applies to backup and restoration for the financial industry and online core transactions that require high data consistency.	In sharding scenarios, global data consistency is ensured.	• Backup and restoration is slow. • Data can be restored by backup set, but cannot be restored by time. • Only 5.3.8 and later support this feature. • During data backup, distributed transactions on instances are locked for seconds. During the locking period, the RT for executing SQL statements may have millisecond-level jitters. Therefore, we recommend that you perform consistent backup during off-peak hours.

13.1.4.6.2. Configure an automatic backup policy

provides the automatic backup feature. This topic describes how to configure an automatic backup policy.

Procedure

1. Log on to the PolarDB-X console.
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More** > **Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Data Recovery** > **Backup and Recovery**.
5. On the page that appears, choose **Backup Policy** > **Edit**.
6. In the **Backup Policy** dialog box, set parameters as needed, and click **OK**.

13.1.4.6.3. Configure local logs

You can use local logs with the backup and recovery feature or the SQL flashback feature of to accurately recover an instance or a database to the desired time point. This topic describes how to configure local logs.

Procedure

1. Log on to the PolarDB-X console.
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More** > **Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Data Recovery** > **Backup and Recovery**.
5. On the page that appears, click the **Local Log Settings** tab and then click **Edit**.
6. In the **Local Binlog Settings** dialog box, set parameters as needed, and click **OK**.

> **Notice** The local log settings are applied to all underlying ApsaraDB RDS for MySQL instances.

13.1.4.6.4. Manual backup

also provides the manual backup capability, so that you can back up data at any time. This topic describes how to manually back up instances and databases.

Procedure

1. Log on to the PolarDB-X console.
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Data Recovery > Backup and Recovery**.
5. On the page that appears, click **Data Backup** on the right.
6. In the dialog box that appears, set Backup Method and Backup Level.
   - Backup Method can be set to **Fast Backup** and **Consistent Backup**. For more information about differences between the two methods, see Backup methods.

     > **Notice** If you select Consistent Backup, distributed transactions are locked within seconds and the response time (RT) may vary by sub-seconds. Therefore, we recommend that you perform this operation during off-peak hours.

   - Backup Level can be set to **Instance Backup** or **Database Backup**. You can select **Instance Backup** to back up the entire instance, or select **Database Backup** to back up a database as needed.
7. Click **OK**.

13.1.4.6.5. Recover data

You can use the data recovery feature of to recover an instance or a database to the time when the backup is created. You can perform this operation at any time. This topic describes how to recover the data of an instance or a database to a specific point in time.

Procedure

1. Log on to the PolarDB-X console.
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Data Recovery > Backup and Recovery**.
5. On the page that appears, click **Data Recovery (Original Clone Instance)** on the right.
6. Select a recovery method
   - **By Time**: Recover data to the selected point in time. You must set **Restoration Time** and **Recovery Level**.
   - **By Backup Set**: Recover data from the selected backup file.

     > **Note** You can also click **Recover** in the Actions column of the target backup set to recover data by backup set

7. Click **Precheck** to check whether a valid backup set is available for data recovery. If the precheck fails, the

data cannot be restored.

8. Click **Enable** to access the order confirmation page.
9. Confirm the order details and then click **Enable** to recover the data. You can view the data recovery progress in **Task Progress** in the upper-right corner of the page.

13.1.4.6.6. SQL flashback

13.1.4.6.6.1. Overview

provides the SQL flashback feature to recover data of particular rows.

When you mistakenly run an SQL statement such as INSERT, UPDATE, or DELETE on , provide the relevant SQL information to match the event in the binary log file and generate the corresponding recovery file. You can download the file and recover data as needed. SQL flashback automatically chooses **fuzzy match** or **exact match** to locate lost data caused by the error. For more information, see Exact match and fuzzy match and Rollback SQL statements and original SQL statements.

Features

- Easy-to-use: SQL flashback allows you to retrieve the lost data by entering required information about the corresponding SQL statement.
- Fast and lightweight: Regardless of the backup policy of ApsaraDB RDS for MySQL instances, you only need to enable log backup before an SQL statement error occurs.
- Flexible recovery: Rollback SQL statements and original SQL statements are available for different scenarios.
- Exact match: SQL flashback supports exact match of data about the corresponding SQL statement, which improves precision of data recovery.

Limits

- SQL flashback depends on the binary log retention time and the log backup feature of ApsaraDB RDS for MySQL must be enabled. Binary log files can be retained only for a certain period. Use SQL flashback to generate files for recovery as soon as possible when an error occurs.
- The recovery files generated by SQL flashback are retained for seven days by default, and you need to download these files as soon as possible.
- The following conditions must be met for SQL flashback exact match:
  - The instance version is 5.3.4-15378085 or later.
  - The version of the ApsaraDB RDS for MySQL instance used by the database is 5.6 or later.
  - SQL flashback exact match is enabled before the error SQL statement is executed.
  - The TRACE_ID information for the error SQL statement is provided.
- To ensure the precision of data recovery, the exact match feature is enabled by default for the database created in a instance of 5.3.4-15378085 or later. After this feature is enabled, SQL execution information is included in the binary log file by default, which requires more storage space for ApsaraDB RDS for MySQL instances. If you need to use the exact match feature, we recommend that you upgrade before enabling the feature. For more information, see Enable exact match.

13.1.4.6.6.2. Generate a recovery file

Procedure

1. Log on to the PolarDB-X console.
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More** > **Manage** from the Actions column of the instance to access the

Basic Information page.

4. In the left-side navigation pane, choose **Data Recovery > SQL Flashback**. The **SQL Flashback** page appears.
5. On the **SQL Flashback** page, enter the basic information about a mistaken SQL statement, including Database, Time Range, Table Name, TRACE_ID, and SQL Statement Type. The following table describes the parameters.

Parameter	Description
Database	The database where the mistaken SQL statement was executed.
Time Range	The time range during which the mistaken SQL statement was executed. The start time is earlier than the start time when the mistaken SQL statement was executed, whereas the end time is later than the time when the execution of the mistaken SQL statement ended. To ensure efficient recovery, we recommend that you limit the time range to five minutes.
Table Name	The name of the table on which the mistaken SQL statement was executed. This parameter is optional.
TRACE_ID	The unique TRACE_ID that allocates for each executed SQL statement. You can obtain the TRACE_ID of the mistaken SQL statement by using the SQL audit feature of .
SQL Statement Type	The type of the mistaken SQL statement. Valid values: ○ INSERT ○ UPDATE ○ DELETE

6. Click **Precheck**. The system checks whether a binary log file exists within the specified time range. For more information about binary log files, see Configure local logs.

> **Note**
> - If no binary log file exists within specified the time range, the precheck fails and the system cannot recover the data for you.
> - If a binary log file exists within the specified time range, the precheck is successful and you can go to the next step.

7. Set SQL Statement Type for Recovery to **Rollback SQL** or **Original SQL Statement**. For more information about differences between the two methods, see Rollback SQL statements and original SQL statements.
8. Click **Generate SQL** to generate an SQL flashback task. The statuses of the SQL flashback tasks that are running on the current instance appear at the bottom of the page.

What's next

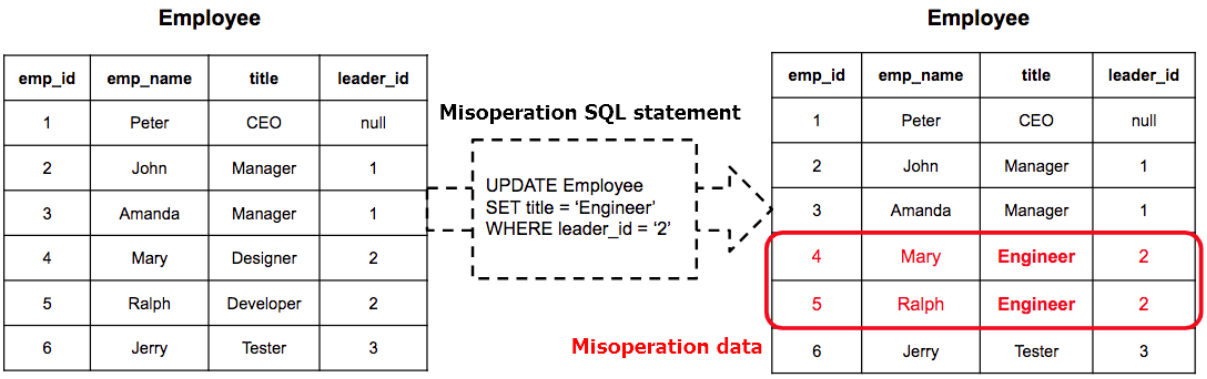
After an SQL flashback task is completed, the task information such as the exact match status and the number of recovered rows appears. You can click **Download** next to the target SQL flashback task to download the corresponding recovery file.

> **Notice** By default, the recovery file is retained for seven days. Download it as soon as possible.

13.1.4.6.6.3. Rollback SQL statements and original SQL statements

To support different business scenarios, SQL flashback provides rollback SQL statements and original SQL statements. Before generating an SQL statement for recovering data, you must select a corresponding recovery method based on your scenario.

Recovery methods



Recovery method	Description	Example
Rollback SQL statement	Traverses the events in the binary log file in reverse order to reverse the INSERT, UPDATE, and DELETE events. • The reverse of INSERT is equivalent to DELETE. • The reverse of DELETE is equivalent to INSERT. • The reverse of UPDATE is equivalent to the value before UPDATE.	UPDATE Employee SET title = 'Developer' WHERE emp_id = '5' UPDATE Employee SET title = 'Designer' WHERE emp_id = '4'
Original SQL statement	Traverses the events in the binary log file in order to mirror all records of the INSERT, UPDATE, and DELETE events. • An INSERT mirror is equivalent to INSERT. • A DELETE mirror is equivalent to INSERT. • An UPDATE mirror is equivalent to the value before INSERT.	INSERT INTO Employee(emp_id,emp_name,title,leader_id) values('4','Mary','Designer','2') INSERT INTO Employee(emp_id,emp_name,title,leader_id) values('5','Ralph','Developer','2')

13.1.4.6.6.4. Exact match and fuzzy match

SQL flashback supports **exact match** and **fuzzy match** for binary log events. You do not need to select a match policy. SQL flashback automatically detects and selects the optimal match policy, and notifies you when the flashback task is completed.

Match mode	Description	Advantage	Disadvantage

Match mode	Description	Advantage	Disadvantage
Exact match	The system performs exact match on the event of a mistaken SQL statement in the binary log file and generates a recovery file.	The recovery file contains only data that is deleted or modified by the mistaken SQL statement. You can use the file directly to ensure the precision and efficiency of data recovery.	The following requirements must be met: • The instance is Version 5.3.4-15378085 or later. • The version of the ApsaraDB RDS for MySQL instance used by the database is Version 5.6 or later. • You have enabled exact match of SQL flashback before the mistaken SQL statement is executed. • You must provide the TRACE_ID of the mistaken SQL statement.
Fuzzy match	The system matches the information about the mistaken SQL statement in the binary log file, including the time range, table name, and SQL statement type. Then, the system generates a recovery file.	Fuzzy match is supported for all instances, regardless of the instance version or parameter settings.	Data that is deleted or modified by the mistaken SQL statement cannot be accurately matched. The recovery file contains data changes made by other business SQL operations. You must filter the required data.

Enable exact match

> **Note** Fuzzy match is enabled by default.

1. Log on to console, and go to the parameter settings page of the specified instance. For more information, see Set parameters.
2. Change the value of **ENABLE_SQL_FLASHBACK_EXACT_MATCH** to **ON**.

13.1.4.6.7. Table recycle bin

13.1.4.6.7.1. Overview

The table recycle bin of allows you to recover mistakenly deleted tables.

After the table recycle bin is enabled for your database, the tables that are deleted by using the DROP TABLE statement are moved to the recycle bin and are no longer visible to you. After the tables are moved to the recycle bin for two hours, they are automatically cleared and cannot be recovered. You can view, recover, and clear the deleted tables in the recycle bin.

Limits and notes

- The table recycle bin feature is only supported by 5.3.3-1670435 and later. For more information, see View the instance version.
- The table recycle bin is disabled for your database by default. For more information about how to enable it,

see Enable the table recycle bin.

- The table recycle bin of does not support the recovery of tables deleted by the TRUNCATE TABLE command.
- Tables in the recycle bin still occupy the storage space of ApsaraDB RDS for MySQL before they are automatically cleared. To release the storage space as soon as possible, you can access the recycle bin to manually delete them.

13.1.4.6.7.2. Enable the table recycle bin

This topic describes how to enable the table recycle bin.

Procedure

1.
2.
3.
4. In the left-side navigation pane, choose **Data Recovery > Table Recycle Bin**. The **Table Recycle Bin** page appears.
5. On the top of the **Table Recycle Bin** page, click the tab of the database for which the table recycle bin needs to be enabled.
6. Click **Enable**.
7. In the dialog box that appears, click **OK**.

13.1.4.6.7.3. Recover tables

This topic describes how to recover your tables from the table recycle bin.

Procedure

1.
2.
3.
4. In the left-side navigation pane, choose **Data Recovery > Table Recycle Bin**. The **Table Recycle Bin** page appears.
5. On the top of the **Table Recycle Bin** page, click the tab of the database in which the tables need to be recovered.
6. Click **Restore** in the **Actions** column of the target table.

13.1.4.6.7.4. Delete tables from the recycle bin

This topic describes how to delete unnecessary tables from the table recycle bin.

Procedure

1.
2.
3.
4. In the left-side navigation pane, choose **Data Recovery > Table Recycle Bin**. The **Table Recycle Bin** page appears.
5. On the top of the **Table Recycle Bin** page, click the tab of the database in which the tables need to be cleared.
6. Click **Delete** in the **Actions** column of the target table.

13.1.4.6.7.5. Disable the table recycle bin

If you no longer need the table recycle bin, you can disable it.

Procedure

1.
2.
3.
4. In the left-side navigation pane, choose **Data Recovery > Table Recycle Bin**. The **Table Recycle Bin** page appears.
5. On the top of the **Table Recycle Bin** page, click the tab of the database for which the table recycle bin needs to be disabled.
6. Click **Disable** to disable the table recycle bin for the database.

13.1.4.7. Set parameters

allows you to set parameters for instances and databases. You can view and modify parameter values in the console based on business needs.

> **Note** Parameters cannot be set for read-only PolaDB-X instances.

Procedure

1. Log on to the PolarDB-X console.
2. On the **DRDS Instance Management** page, find the target instance.
3. Click the instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Diagnostics and Optimization > Parameter Settings**. Click the Instance or Database tab to view parameters that you can modify for instances and databases, respectively. For more information about the parameters, see Parameter description.
5. Click next to the parameter you want to modify, enter the target value, and click **OK**.
6. Click **Submit** in the upper-right corner to commit the modification.

   > **Note** To undo parameter modification, click **Cancel** in the upper-right corner.

Parameter description

Parameter	Level	Description
Slow SQL threshold	Instance	The threshold for slow SQL statements. SQL statements whose thresholds exceed this threshold are recorded in logical slow SQL logs.
Logical idle link timeout	Instance	The logical timeout period of the idle connection between user applications and (unit: ms).
Maximum package size	Instance	The maximum network packet for the interaction between user applications and (unit: byte).

Parameter	Level	Description
Instance memory pool size limit	Instance	The maximum size of the memory pool for an instance. If the memory usage on an instance exceeds the value, an error is reported and the query ends.
Whether to prohibit all table deletion/update	Database	Specifies whether to disable full table deletion or update.
Whether to open the recycle bin	Database	Specifies whether to enable the recycle bin for storing deleted logical tables.
Temporary table size	Database	The size of the temporary table used during distributed queries in (unit: row).
Number of join tables	Database	The maximum number of table shards that can be combined through JOIN when you query multiple table shards in a database.
Physical SQL timeout	Database	The timeout period of SQL statements for interaction between and ApsaraDB RDS for MySQL (unit: ms). The value 0 indicates the timeout period is not limited.
SQL exact flashback switch	Database	Specifies whether to support SQL flashback exact match. It is disabled by default. After it is enabled, information about the queries is added to the binary log file used by the database.
Whether to enable logical INFORMATION_SCHEMA query	Database	Specifies whether to enable logical INFORMATION_SCHEMA query (not relying on the shadow database but returning the aggregation results of logical databases and tables). When it is disabled, the original status is restored (relying on the shadow database and returning the physical database and table information).
Transaction log cleanup start time period	Database	The period during which transaction log cleanup starts at a random time.
Library-level memory pool size limit	Database	The maximum size of the database-level memory pool. When the memory usage of a database exceeds this value, an error is reported and the query terminates. The value -1 indicates no limit.
Query-level memory pool size limit	Database	The maximum size of the query-level memory pool. When the memory usage of a query exceeds this value, an error is reported and the query terminates. The value -1 indicates no limit.
Whether CBO is enabled	Database	Specifies whether to enable the cost-based optimizer (CBO), including features such as Join Reorder and Hash Join.
Whether to enable the asynchronous DDL engine	Database	Specifies whether to enable the data definition language (DDL) engine. If you disable it, the execution logic of the original DDL engine remains.

Parameter	Level	Description
Whether to enable asynchronous-only mode under asynchronous DDL engine	Database	Specifies whether to enable the asynchronous-only mode when the asynchronous DDL engine is enabled. • Enabled: The status is returned immediately after the client connects to and executes the DDL statement. The execution status can be viewed only through asynchronous DDL management statements. • Disable: The synchronous mode remains. That is, the status is returned only after the client completes executing the DDL statement.
Maximum number of physical tables allowed to be created in a single physical database	Database	The maximum number of table shards that can be created in a database shard.
INFORMATION_SCHEMA.TABLES queries whether statistics are aggregated	Database	Specifies whether to aggregate statistics of INFORMATION_SCHEMA.TABLES queries. To ensure the performance, it is not aggregated by default.
Maximum number of physical sharding links	Database	The maximum number of connections between and a single ApsaraDB RDS for MySQL shard.
Minimum number of physical sharding links	Database	The minimum number of connections between and a single ApsaraDB RDS for MySQL shard.
Physical idle link timeout	Database	The idle time of the connection between and ApsaraDB RDS for MySQL (unit: minute).

13.1.4.8. SQL audit and analysis

13.1.4.8.1. Description

Cloud Native Distributed Database PolarDB-X (PolarDB-X) combines the SQL audit and analysis feature with Log Service (SLS). This feature not only audits historical SQL records, but also provides real-time diagnosis and analysis of SQL execution status, performance metrics, and security risks. You can enable SQL audit and analysis in the PolarDB-X console.

Benefits

- Easy operation: SQL audit and analysis can be enabled with easy configuration to help you audit and analyze SQL logs in real time.
- Lossless performance: Pulling SQL log files from PolarDB-X nodes and uploading these logs to SLS in real time does not affect instance performance.
- Trace to historical issues: This feature supports importing historical SQL logs to trace issues.
- Real-time analysis: This feature provides real-time SQL analysis and an out-of-the-box report center based on SLS. This feature also supports custom reports and drill-down analysis, and helps you understand the execution status, performance, and security risks of databases.
- Real-time alerts: This feature supports real-time monitoring and alerts based on customized metrics to ensure timely response to critical business exceptions.

Limits and instructions

- You must activate Alibaba Cloud SLS to use the SQL audit and analysis feature.
- SQL audit logs are saved for 30 days by default. You can modify the log storage time as needed.
- Do not delete or modify the default settings for the project, Logstore, index, or dashboard that are created by SLS. SLS updates and upgrades the SQL log audit feature from time to time. The indexes and default reports of the exclusive Logstore are also automatically updated.
- The maximum length of a single SQL statement is 5 MB.

Scenarios

- Troubleshoot SQL problems

  After the SQL audit and analysis feature is enabled, you can quickly search SQL logs to locate and troubleshoot problems. For example, to check whether a DROP operation is performed, you can perform the following query:

  ```
  sql_type: Drop
  ```

  The query result contains information such as the SQL execution time, user, and IP address of the client that runs the SQL statement.

- Analyze costly SQL templates

  In most applications, SQL statements are dynamically generated based on several templates, with different parameters. The real-time analysis feature of SLS allows you to obtain the list of costly SQL statements in the current database.

  For example, execute the following query:

  ```
  | SELECT sql_code as "SQL template ID",
  round(total_time * 1.0 /sum(total_time) over() * 100,2) as "execution time share (%)",
  execute_times as "number of execution times",
  round(avg_time) as "average execution time",
  round(avg_rows) as "average number of affected rows",
  CASE WHEN length(sql) > 200 THEN  concat(substr(sql, 1, 200), '......') ELSE trim(lpad(sql, 200,$)
  end as "sample SQL" FROM (SELECT sql_code, count(1) as execute_times,
  sum(response_time) as total_time,
  avg(response_time) as avg_time,
  avg(affect_rows) as avg_rows,
  arbitrary(sql) as sql FROM log GROUP BY sql_code) ORDER BY "execution time share (%)" desc limit 1
  0
  ```

  The search result contains the SQL template ID, ratio of response time of the statement generated from the template in the total response time of SQL statements, number of executions, average execution time, average number of affected rows, and sample SQL statement. You can find and optimize the most costly SQL templates in the application based on the analysis result.

- Collect log statistics

  To help you analyze issues, PolarDB-X combines the SQL audit and analysis feature with SLS and provides out-of-the-box reports. You can diagnose and analyze the running status, performance, and potential security risks of databases in real time.

13.1.4.8.2. Enable SQL audit and analysis

The SQL audit and analysis feature is disabled by default. You can manually enable it in the console. By default, you can perform only audit and analysis on the log data generated after the SQL audit and analysis feature is enabled. You can also import a portion of historical data.

Prerequisites

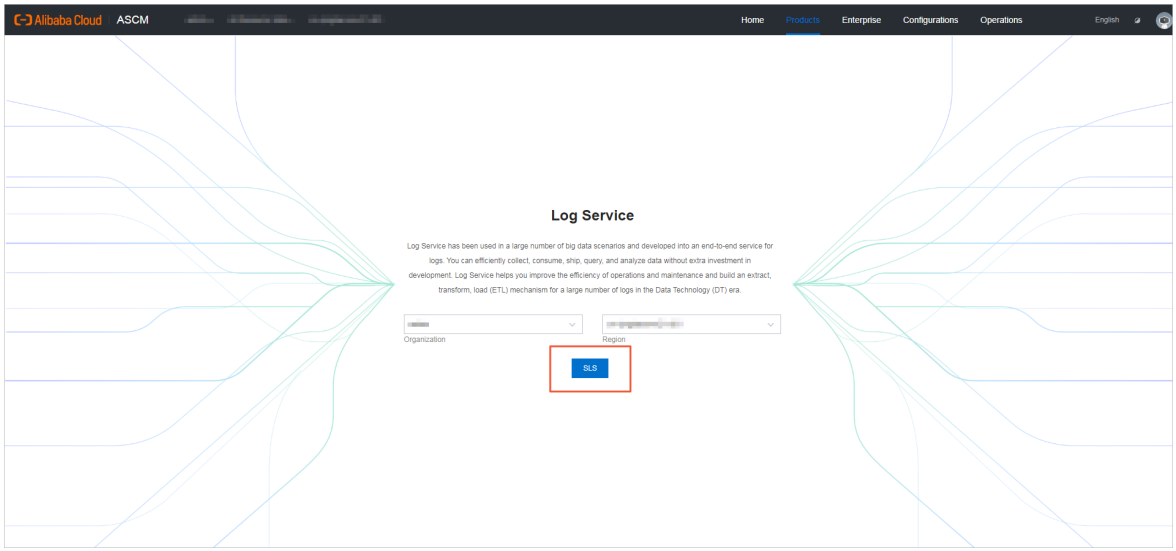
The SQL audit and analysis feature depends on Log Service (SLS). You must activate SLS before you use this feature.

Procedure

1. Log on to the SLS console. For more information, see *Log Service*User Guide > Quick Start > Log on to the Log Service console.
2. Select the organization to which the instance belongs.

   > **Note** The logon account must be consistent with the logon account of .

3. Click **SLS** to go to the Log Service page.



4. Log on to the PolarDB-X console.
5. Find the target instance in the instance list.
6. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
7. In the left-side navigation pane, choose **Diagnostics and Optimization > SQL Audit and Analysis**.
8. In the left-side section, select the database for which you want to enable the SQL audit and analysis feature.
9. On the SQL Audit and Analysis page, turn on the switch next to **SQL Audit Log Status of Current Database** on the right.

   > **Note** On the SQL Audit and Analysis page, you can also turn on the switch of **SQL Audit and Analysis** next to the target database in the left-side section.

10. Confirm whether to import historical data.

    > **Note** By default, you can analyze and audit only the logs that are generated after the SQL audit and analysis feature is enabled. If you find the historical data of the database is modified but the SQL audit and analysis feature is not enabled, you can import historical data and include historical logs in the audit and analysis scope to trace data tampering. dynamically checks the scope of historical data that can be imported based on the log storage on the instance. Logs within seven days can be imported.
    >
    > - If you need to import historical data, enable **Import Historical Data or Not**, specify the backtrace start time and end time, and then click **Enable**.
    > - If you do not need to import historical data, click **Enable**.

What's next

Every time you use the SQL audit and analysis feature, you must repeat the preceding steps.

13.1.4.8.3. Log fields

This topic describes the log fields in SQL audit and analysis.

Field	Description	Supported version
__topic__	The log topic in the format of `drds_audit_log_{instance_id name_{ db_name}` , such as `drds_audit_log_drdsxyzabcd_demo_drds_db` .	All versions
instance_id	The ID of the instance.	All versions
db_name	The name of the database	All versions
user	The user name used to run the SQL statement.	All versions
client_ip	The IP address of the client that accessed the instance.	All versions
client_port	The port of the client that accessed the instance.	All versions
sql	The executed SQL statement.	All versions
trace_id	The trace ID of the SQL statement when it was executed. If a transaction was executed, it is tracked by an ID that consists of the trace ID, a hyphen, and a number, for example, `drdsabcdxyz-1` and `drdsabcdxyz-2` .	All versions
sql_code	The hash value of the template SQL statement.	All versions
hint	The hint that was used to execute the SQL statement.	All versions
table_name	The name of the table involved in the query. Separate multiple tables by commas (,).	All versions
sql_type	The type of the SQL statement. Valid values: SELECT, INSERT, UPDATE, DELETE, SET, ALTER, CREATE, DROP, TRUNCATE, REPLACE, and Other.	All versions
sql_type_detail	The name of the SQL parser.	All versions
sql_time	The start time for the execution of the SQL statement. The time follows the `yyyy-MM-dd HH:mm:ss.SSS` format.	All versions

Field	Description	Supported version
response_time	The response time. Unit: milliseconds.	Version 5.3.4-15378085 and later
affect_rows	The number of rows returned when the SQL statement was executed. The number of rows affected when the INSERT, DELETE, or UPDATE statement was executed.	Version 5.3.4-15378085 and later
fail	Indicates whether an error occurred in the execution of the SQL statement. Valid values: • 0: successful • 1: failed	Version 5.3.4-15378085 and later

13.1.4.8.4. Log analysis

The SQL audit and analysis feature is based on Log Service (SLS) and provides powerful log analytics capabilities. This topic describes SQL statements for log analysis in common scenarios and provides relevant examples.

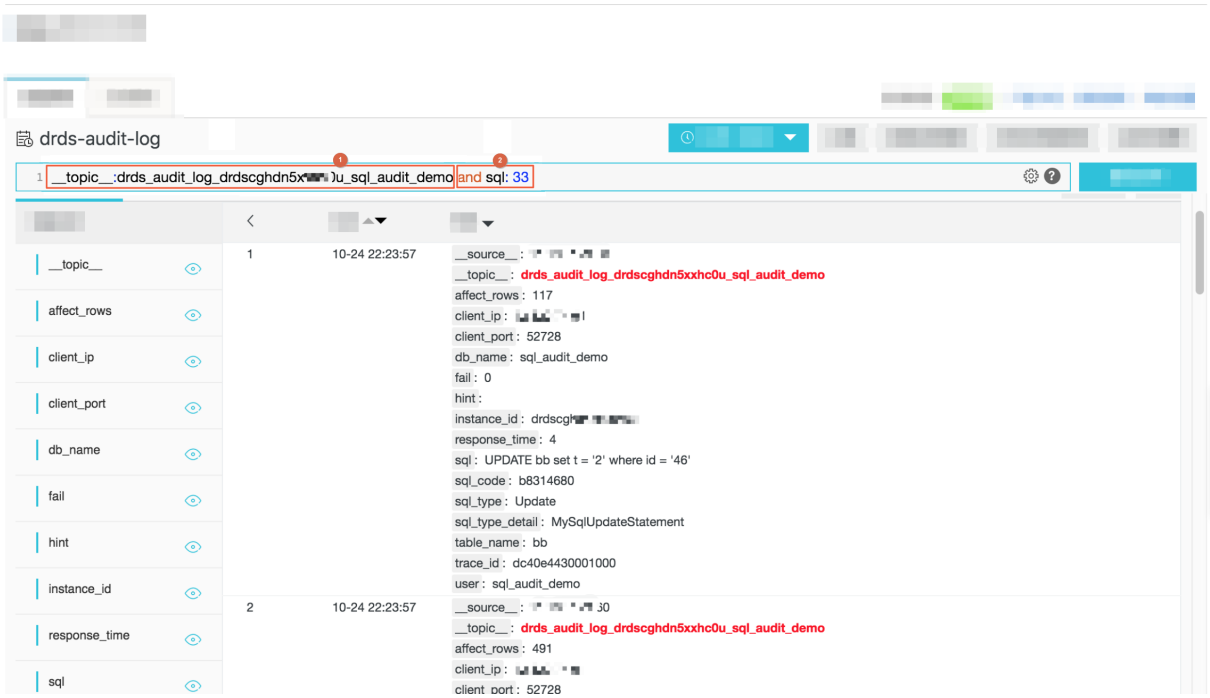
After the SQL audit and analysis feature is enabled, you can perform audit and analysis on SQL log files by using the query and analysis syntax of SLS on the SQL Audit and Analysis page. Based on the query and analysis syntax of SLS, you can find problematic SQL statements on the Log Analysis tab and analyze the SQL statement execution status, performance metrics, and security issues of . For more information about the query and analysis syntax of SLS, see *Log Service User Guide > Query and Analysis > Query Syntax and Functions > Query Syntax*.

Precautions

All the audit logs of databases in the same region are written to the same Logstore in SLS. Therefore, by default, the SQL Audit and Analysis page provides the filter conditions based on __topic__, to ensure that the searched SQL log files are from . Therefore, all the statements provided in this topic must be used after the existing filter conditions.

An example is shown in the following figure:

- The ① part is the default filter condition.
- The ② part is the additional filter condition.



Find problematic SQL statements

- Fuzzy search

  For example, to query SQL statements that contain the "34" keyword, enter the following content in the search box:

  ```
  and sql: 34
  ```

  The result is shown in the following figure.

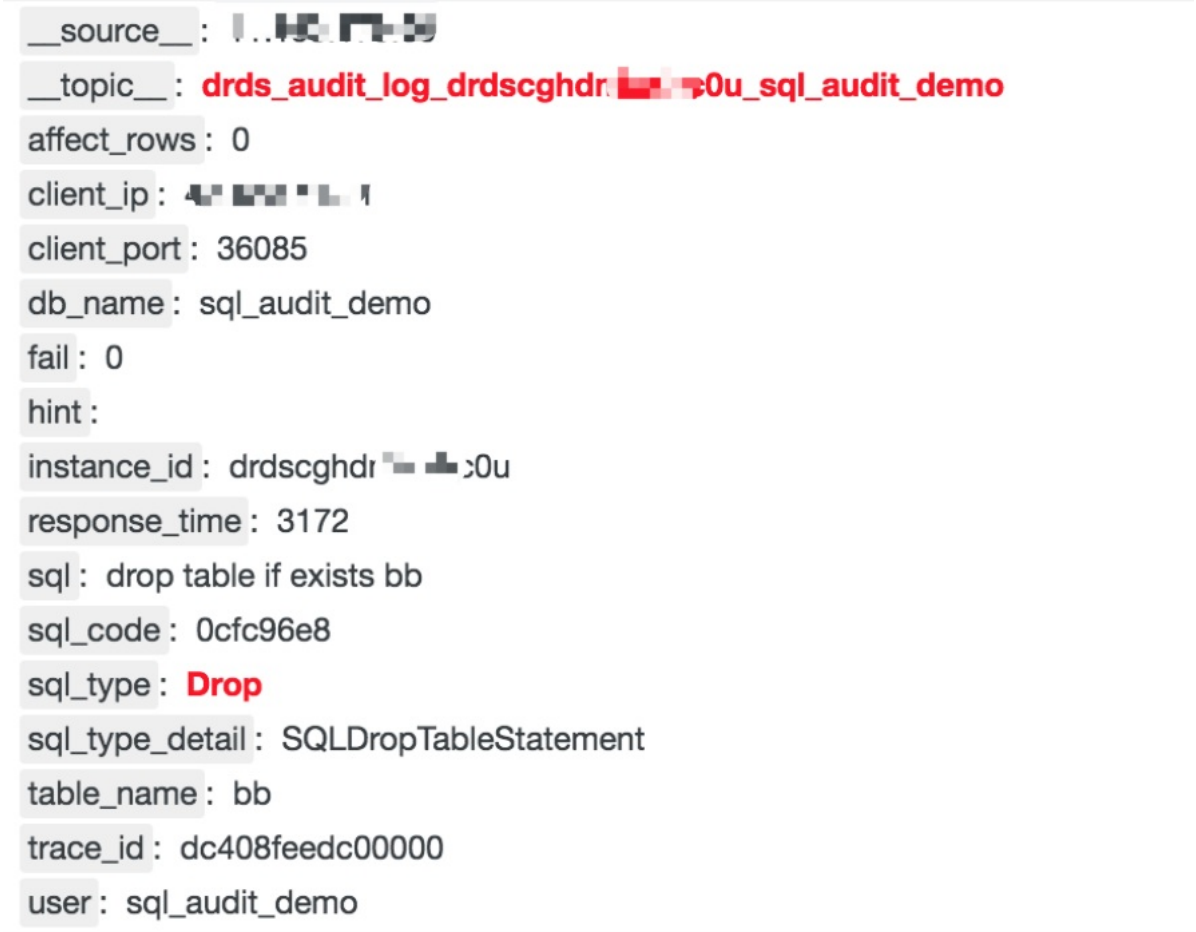
- Field search

Based on built-in index fields, the SQL audit and analysis feature also supports field-based search.

For example, to query SQL statements of the Drop type, execute the following statement:

```
and sql_type:Drop
```

The result is shown in the following figure.



- Multi-condition search

  You can use the "and" and "or" keywords to perform a multi-condition search.

  For example, you can query the delete operation on the rows whose id is 34:

  ```
  and sql:34 and sql_type: Delete
  ```

- Search based on numeric comparison

  affect_rows and response_time in the index filed are numeric values and support comparison operators.

  For example, you can query the SQL INSERT statements whose response_time is greater than 1s.

  ```
  and response_time  > 1507 and sql_type: Insert
  ```

  For example, you can query the SQL statement that deletes more than 100 rows of data:

  ```
  and affect_rows  > 100 and sql_type: Delete
  ```

Analysis of the SQL statement execution status

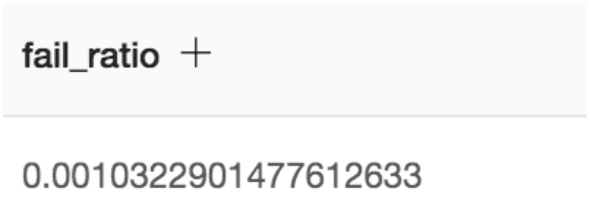
This section introduces the statements used to query the SQL statement execution status in .

- Failure rate of SQL statement execution

  Execute the following statement to query the failure rate of SQL statement execution:

  ```
  | SELECT sum(case when fail = 1 then 1 else 0 end) * 1.0 / count(1) as fail_ratio
  ```

  The result is shown in the following figure.

  If your business is sensitive to the error rate of SQL statement execution, you can customize the alert information based on the query result. Click **Save as Alert** in the upper-right corner of the page.

  In the alert settings shown in the preceding figure, the number of log entries that have an error rate of SQL statement execution greater than 0.01 within 15 minutes is checked within every 15 minutes. You can also customize alerts as needed.

- Total number of rows returned by SELECT statements

  Execute the following statement to query the cumulative number of rows queried by SELECT statements:

  ```
  and sql_type: Select | SELECT sum(affect_rows)
  ```

- SQL statement type distribution

  Execute the following statement to query the SQL statement type distribution:

  ```
  | SELECT  sql_type, count(sql) as times GROUP BY sql_type
  ```

- IP address distribution of SQL independent users

  Execute the following statement to query the distribution of IP addresses of independent users who execute SQL statements:

  ```
  | SELECT  user, client_ip, count(sql) as times GROUP BY user, client_ip
  ```

SQL performance analysis

This section describes typical SQL statements for SQL performance analysis.

- Average response time of SELECT statements

  Execute the following statement to query the average response time of SELECT statements:

  ```
  and sql_type: Select | SELECT avg(response_time)
  ```

- Distribution of SQL statement response time

  Execute the following statement to query the distribution of SQL statement response time:

  ```
  and response_time > 0 | select case when response_time <=10 then '<=10 ms when response_time > 10 and response_time <= 100 then '10~100 ms when response_time > 100 and response_time <= 1000 then '100 ms ~ 1s 'When response_time > 1000 and response_time <= 10000 then '1s ~ 10s' when response_time > 10000 and response_time <= 60000 then '10s ~ 1 min'> 1 min' end as latency_type, count(1) as cnt group by latency_type order by latency_type DESC
  ```

  The preceding query shows the distribution of SQL statement execution time based on a given time range. You can adjust the time range to obtain finer-grained results.

- Top 50 slow SQL statements

Execute the following statement to query slow SQL statements:

```
 | SELECT date_format(from_unixtime(__time__), '%m/%d %H:%i:%s') as time, user, client_ip, client_port, sql_type, affect_rows, response_time, sql ORDER BY  response_time desc LIMIT 50
```

The following figure shows the result, which includes the SQL statement execution time, user name, IP address, port number, SQL statement type, number of affected rows, response time, and text of SQL statements.

time	user	client_ip	client_port	sql_type	affect_rows	response_time	sql
09/28 14:04:05	sql_audit_demo	[redacted]	[redacted]7	Drop	0	9583	drop table if exists bb
09/28 14:04:05	sql_audit_demo	[redacted]	[redacted]7	Drop	0	9583	drop table if exists bb
09/28 14:04:05	sql_audit_demo	[redacted]	[redacted]477	Drop	0	9583	drop table if exists bb
09/27 17:38:18	sql_audit_demo	[redacted]	[redacted]0073	Drop	0	7200	drop table if exists bb

- Top 10 costly SQL templates

  In most applications, SQL statements are dynamically generated based on several templates, and only the parameters are different. You can find, analyze, and optimize the costly SQL templates based on template IDs. Enter the following query statement:

  ```
  | SELECT sql_code as "SQL template ID", round(total_time * 1.0 /sum(total_time) over() * 100, 2) as "response time share (%)" ,execute_times as "number of executions", round(avg_time) as "average response time", round(avg_rows) as "average number of affected rows", CASE WHEN length(sql) > 200 THEN concat(substr(sql, 1, 200℃, '......') ELSE trim(lpad(sql, 200,'hour) end as "sample SQL" FROM (SELECT sql_code, count(1) as execute_times, sum(response_time) as total_time, avg(response_time) as avg_time, avg(affect_rows) as avg_rows, arbitrary(sql) as sql FROM log GROUP BY sql_code) ORDER BY "execution time share (%)" desc limit 10
  ```

  The statistics include the SQL template ID, percentage of response time of the statement generated from the template in the total response time of SQL statements, number of executions, average response time, average number of affected rows, and sample SQL statement. For better display effect, each page displays 200 entries. In the preceding query result, statements are ranked by the response time share. However, you can rank the statements by the average response time or the number of executions to troubleshoot relevant issues.

- Average transaction response time

  For SQL statements within the same transaction, the preset trace_id field prefixes are the same, and the suffixes are '-' followed by sequence numbers. trace_id of non-transactional SQL statements does not contain '-'. Based on this, you can analyze the performance of transactions.

  > **Note** Transaction analysis is less efficient than other query operations because it involves prefix matching.

  For example, execute the following statement to query the average response time of transactions:

  ```
   | SELECT  sum(response_time) / COUNT(DISTINCT substr(trace_id, 1, strpos(trace_id, '-') - 1)) where strpos(trace_id, '-') > 0
  ```

- Top 10 slow transactions

  You can query the list of slow transactions by response time of transactions. Use the following statement:

  ```
  | SELECT substr(trace_id, 1, strpos(trace_id, '-') - 1) as "transaction ID" , sum(response_time) as "response time" where strpos(trace_id, '-') > 0 GROUP BY substr(trace_id, 1, strpos(trace_id, '-') - 1) ORDER BY "response time" DESC LIMIT 10
  ```

Based on this, you can use the transaction ID to search for all the SQL statements under the transaction and analyze the specific causes of slow execution. Use the following statement:

```
and trace_id: db3226a20402000*
```

- Top 10 transactions with batch operations

  Based on the number of rows affected by SQL statements in a transaction, you can obtain the list of transactions that contain batch operations. Use the following statement:

```
| SELECT substr(trace_id, 1, strpos(trace_id, '-') - 1) as "transaction ID" , sum(affect_rows) as "number of affected rows" where strpos(trace_id, '-') > 0 GROUP BY substr(trace_id, 1, strpos(trace_id, '-') - 1) ORDER BY "number of affected rows" DESC LIMIT 10
```

SQL security analysis

This section provides typical query statements for SQL security analysis.

- Distribution of types of failed SQL statements

```
and fail > 0 | select sql_type, count(1) as "number of errors" group by sql_type
```

- High-risk SQL statements

  High-risk SQL statements are of the Drop or Truncate type. You can also add more conditions as needed.

```
and sql_type: Drop OR  sql_type: Truncate
```

- SQL batch delete events

```
 and affect_rows > 100 and sql_type: Delete | SELECT date_format(from_unixtime(__time__), '%m/%d %H:%i:%s') as time, user, client_ip, client_port, affect_rows, sql ORDER BY  affect_rows desc LIMIT 50
```

13.1.4.8.5. Log reports

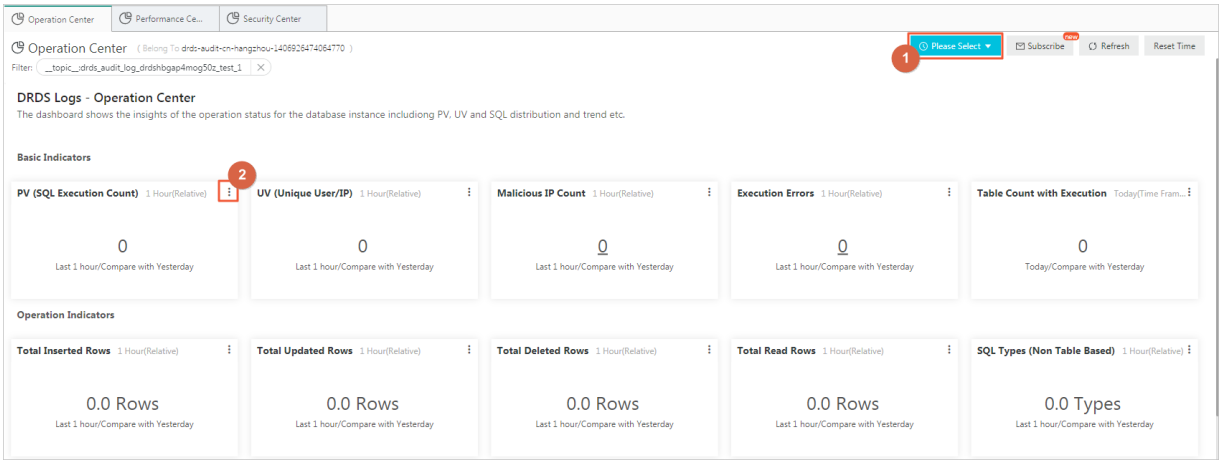
Based on Log Service (SLS), the SQL audit and analysis feature of provides out-of-the-box report centers, including the Operation Center, Performance Center, and Security Center. This feature allows you to fully understand the performance status, performance metrics, and potential security risks of your databases.

After Enable SQL audit and analysis, click the **Log Reports** tab on the current page. You can view the reports pages provided by SLS, including **Operation Center**, **Performance Center**, and **Security Center**.

> **Note**
>
> - All the audit logs of databases in the same region are written to the same Logstore in SLS. Therefore, when you view the reports of the current database, filter conditions based on the __topic__:drds_audit_log_ instance id_database name are added by default, which indicates that you are viewing the data of the current database. For example, `drds_audit_log_drdsxyzabcd_demo_drds_db` .
> - If the version of the instance is earlier than Version 5.3.4-15378085, the relevant fields are missing from SQL logs. For more information about log fields, see Log fields. The Log Reports tab provides only a simplified version of Operation Center. To use a full version of reports, upgrade the instance to the latest version.

View reports

Statistics in the charts on the Log Reports tab are generated for different time periods. You can change the time range as needed. You can change the time range for all charts or a single chart.



- Click the time selector (position ① in the figure). In the dialog box that appears, you can change the time range of all the charts on the current page.
- Click the time selector of a chart (position ② in the figure) to modify the time range of the chart.

Operation Center

Operation Center shows the metrics, distribution, and trends of SQL statement execution in databases.

Item	Type	Default time range	Description
PV (SQL Execution Count)	Single value	1 Hour (Relative)	The number of SQL statement executions
UV (Unique User/IP)	Single value	1 Hour (Relative)	The number of unique user-IP groups
Malicious IP Count	Single value	1 Hour (Relative)	The number of malicious IP addresses. For more information about the definition of malicious IP addresses, see security detection functions.
Execution Errors	Single value	1 Hour (Relative)	The number of SQL statements with execution errors
Table Count with Execution	Single value	1 Hour (Relative)	The total number of tables operated by SQL statements
Total Inserted Rows	Single value	1 Hour (Relative)	The total number of rows inserted by INSERT statements
Total Updated Rows	Single value	1 Hour (Relative)	The total number of rows updated by UPDATE statements
Total Deleted Rows	Single value	1 Hour (Relative)	The total number of rows deleted by DELETE statements

Item	Type	Default time range	Description
Total Read Rows	Single value	1 Hour (Relative)	The total number of rows returned by SELECT statements
SQL Types (Non Table Based)	Single value	1 Hour (Relative)	The types of SQL statements used for non-table operations, such as SHOW VARIABLES LIKE
SQL Execution Trend	Column chart	1 Hour (Relative)	The distribution trend of SQL statement executions and the distribution trend of failed SQL statements
Operated Tables	Flow diagram	1 Hour (Relative)	The distribution of tables operated by SQL statements
SQL Type	Flow diagram	1 Hour (Relative)	The distribution of SQL statement types by time
User Distribution	Pie chart	1 Hour (Relative)	The distribution of users who execute SQL statements
SQL Type Distribution	Area chart	1 Hour (Relative)	The percentage of SQL statement types in the current time range
Tables with Most Operations (Top 50)	Table	1 Hour (Relative)	The list of top tables by the number of operations, including table names and the number of operations such as read, delete, update, and insert
SQL Type (World)	Map	1 Hour (Relative)	The distribution of IP addresses of clients that execute the SQL statements, on the world map
SQL Type (China)	Map	1 Hour (Relative)	The distribution of IP addresses of clients that execute the SQL statements, on the map of China

Performance Center

Performance Center shows performance metrics, the distribution of slow and fast SQL statements, and the distribution and sources of costly SQL statements in databases.

Item	Data type	Default time range	Description
Peak SQL Execution Traffic	Single value	1 Hour (Relative)	The maximum number of SQL statements executed per second

Item	Data type	Default time range	Description
Peak Select Traffic	Single value	1 Hour (Relative)	The maximum number of rows returned by SELECT statements per second
Peak Insert Traffic	Single value	1 Hour (Relative)	The maximum number of rows inserted by INSERT statements per second
Peak Update Traffic	Single value	1 Hour (Relative)	The maximum number of rows updated by UPDATE statements per second
Peak Delete Traffic	Single value	1 Hour (Relative)	The maximum number of rows deleted by DELETE statements per second
Average Response Time	Single value	1 Hour (Relative)	The average response time of SQL statements
Select SQL	Single value	1 Hour (Relative)	The average number of SELECT statements executed per second
Insert SQL	Single value	1 Hour (Relative)	The average number of INSERT statements executed per second
Update SQL	Single value	1 Hour (Relative)	The average number of UPDATE statements executed per second
Delete SQL	Single value	1 Hour (Relative)	The average number of DELETE statements executed per second
Select/Update Traffic Trend	Line chart	1 Hour (Relative)	The distribution of rows affected by the SELECT and UPDATE statements over time
SQL Execution Time Distribution	Pie chart	1 Hour (Relative)	The distribution of execution time of SQL statements
Slow SQL Table Distribution	Pie chart	1 Hour (Relative)	The distribution of tables targeted by slow SQL statements whose response time exceeds 1s
Slow SQL User Distribution	Pie chart	1 Hour (Relative)	The distribution of users who execute slow SQL statements with response time of more than 1s
Slow SQL Type Distribution	Pie chart	1 Hour (Relative)	The distribution of types of slow SQL statements whose response time exceeds 1s

Item	Data type	Default time range	Description
Slow SQL (Top 50)	Table	1 Hour (Relative)	The table of slow SQL statements whose response time exceeds 1s, including the time, client, response time, instance, database, table, user, affected rows, SQL type, and SQL text
SQL Template Execution Time Top 20	Table	1 Hour (Relative)	Statistics of the execution status of the SQL statements in the template based on the specified SQL template, including the SQL template ID, response time share, number of executions, average response time, average number of affected rows, and sample SQL statement
Transaction Affected Rows Top 20	Table	1 Hour (Relative)	The table of top 20 transaction-by the number of affected rows, including the transaction ID and the number of affected rows
Transaction Executed Time Top 20	Table	1 Hour (Relative)	The table of top 20 transactions by response time, including the transaction ID and the number of affected rows

Security Center

Security Center shows failed and malicious SQL statement executions in databases, and the details, distribution, and trends of malicious SQL batch delete and update events.

Item	Type	Default time range	Description
Error Count	Single value	1 Hour (Relative)	The number of failed SQL statement executions
Batch Delete Events	Single value	1 Hour (Relative)	The number of SQL statements for batch delete events (more than 100 rows)
Batch Update Events	Single value	1 Hour (Relative)	The number of SQL statements for batch update events (more than 100 rows)
Malicious SQL Executions	Single value	1 Hour (Relative)	The number of malicious SQL statement executions (Drop and Truncate)

Item	Type	Default time range	Description
Malicious IP Count	Single value	1 Hour (Relative)	The number of malicious IP addresses. For more information about the definition of malicious IP addresses, see security detection functions.
Error Distribution	Area chart	1 Hour (Relative)	The distribution of types of failed SQL statements
Distribution of Client with Errors	Map	1 Hour (Relative)	The distribution of clients for failed SQL statements on the map of China
Client with Most Errors	Table	1 Hour (Relative)	The table of clients on which the execution of SQL statements failed, including the IP address, number of errors, type of failed SQL statement, and sample failed SQL statement
Malicious SQL Executions	Table	1 Hour (Relative)	The table of malicious SQL statement executions, including the time, IP address, SQL, instance ID, database, table, and user
Batch Delete Events (Top 50)	Table	1 Hour (Relative)	The table of top SQL batch delete events, including the earliest execution time, most recent execution time, instance ID, database, table, number of executions, average number of deleted rows, average response time, and sample SQL statement
Batch Update Events (Top 50)	Table	1 Hour (Relative)	The table of top SQL batch update events, including the earliest execution time, most recent execution time, instance ID, database, table, number of executions, average number of updated rows, average response time, and sample SQL statement

13.1.4.9. Monitor PolarDB-X instances

13.1.4.9.1. View monitoring information

provides multi-dimensional monitoring. This topic describes how to view monitoring information in the console.

Procedure

1.
2.
3.
4. On the **Basic Information** page, choose **Monitoring and Alerts > Instance Monitoring** from the left-side navigation pane.
5. On the **Instance Monitoring** page, select a monitoring dimension and the corresponding metrics to view details. For more information about monitoring metrics, see Monitoring metrics.

13.1.4.9.2. Monitoring metrics

Instance monitoring is divided into resource monitoring and engine monitoring. Engine monitoring metrics are classified into metrics at the instance level and metrics at the database level. When some engine monitoring metrics are abnormal, you can directly check the metrics of each database to locate the database with performance problems. The following table describes the metrics of these two types in details.

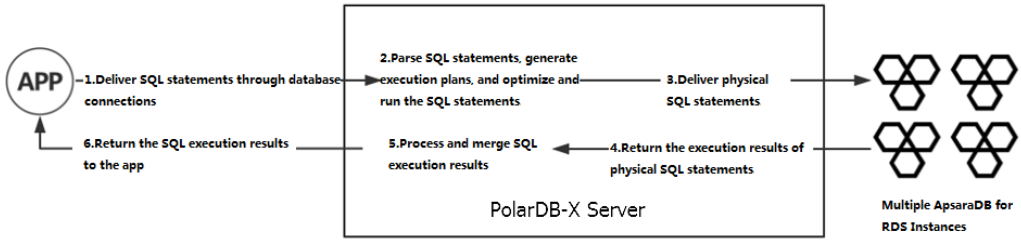
Monitoring item	Category	Description	Data collection cycle	Data retention period	Description
CPU Utilization (%)	Resource monitoring	The average CPU utilization of server nodes.	1 minute	3 days	-
Memory Usage (%)	Resource monitoring	The memory usage of JVM Old Generation on server nodes.	1 minute	3 days	Memory usage fluctuations are normal.
Inbound Traffic (Kbps)	Resource monitoring	The total inbound network traffic of server nodes.	1 minute	3 days	Inbound network traffic is generated when ApsaraDB RDS for MySQL returns data to .
Outbound Traffic (Kbps)	Resource monitoring	The total outbound network traffic of server nodes.	1 minute	3 days	Outbound network traffic is generated when a instance sends a physical SQL statement to an AparaDB RDS for MySQL instance or a instance returns data to an application.

Monitoring item	Category	Description	Data collection cycle	Data retention period	Description
Logical QPS	Engine monitoring	The total number of SQL statements processed per second on server nodes.	5 seconds	7 days	-
Physical QPS	Engine monitoring	The total number of SQL operations sent from server nodes to ApsaraDB RDS for MySQL per second.	5 seconds	7 days	One logical SQL statement can be partitioned into multiple physical SQL statements.
Logical RT (ms)	Engine monitoring	The average response time (RT) for processing each SQL statement by .	5 seconds	7 days	If a logical SQL statement is partitioned into physical SQL statements for delivery, the logical RT of the SQL statement contains the RT of the physical SQL statements.
Physical RT (ms)	Engine monitoring	The average RT for transmitting SQL statements from to ApsaraDB RDS for MySQL.	5 seconds	7 days	-
Connections	Engine monitoring	The total number of connections established between an application and .	5 seconds	7 days	The connections from to ApsaraDB RDS for MySQL are not included.
Active Threads	Engine monitoring	The number of threads that are used by to run SQL statements.	5 seconds	7 days	-

13.1.4.9.3. How metrics work

Before analyzing metrics, you need to understand the execution process of SQL statements on .

SQL execution flowchart



In the entire SQL execution process, the execution status of steps 2 through 4 is reflected in various metrics of .

- In step 2, SQL parsing, optimization, and execution consume CPU resources. A more complex SQL statement (with a complex structure or ultra-long length) consumes more CPU resources. You can run the **TRACE** command to trace the SQL execution process. You can see the time consumed by an SQL statement during optimization. The longer time consumed indicates a higher CPU utilization.
- In step 3, the delivery and execution of physical SQL statements consume I/O resources. You can analyze the execution status of physical SQL statements based on metrics such as logical queries per second (QPS), physical QPS, logical response time (RT), and physical RT. For example, if the physical QPS is low and the physical RT is high, the current ApsaraDB RDS for MySQL instance is processing SQL statements very slowly. You need to check the performance of the ApsaraDB RDS for MySQL instance.
- In step 5, the SQL execution results are processed and integrated. These operations convert the execution results of physical SQL statements. In most cases, only SQL metadata is converted, which consumes few resources. However, the CPU utilization is high for steps such as `heap sort`. For more information about how to determine the consumption of SQL statements at this stage, see Details about a low SQL statement.

13.1.4.9.4. Prevent performance problems

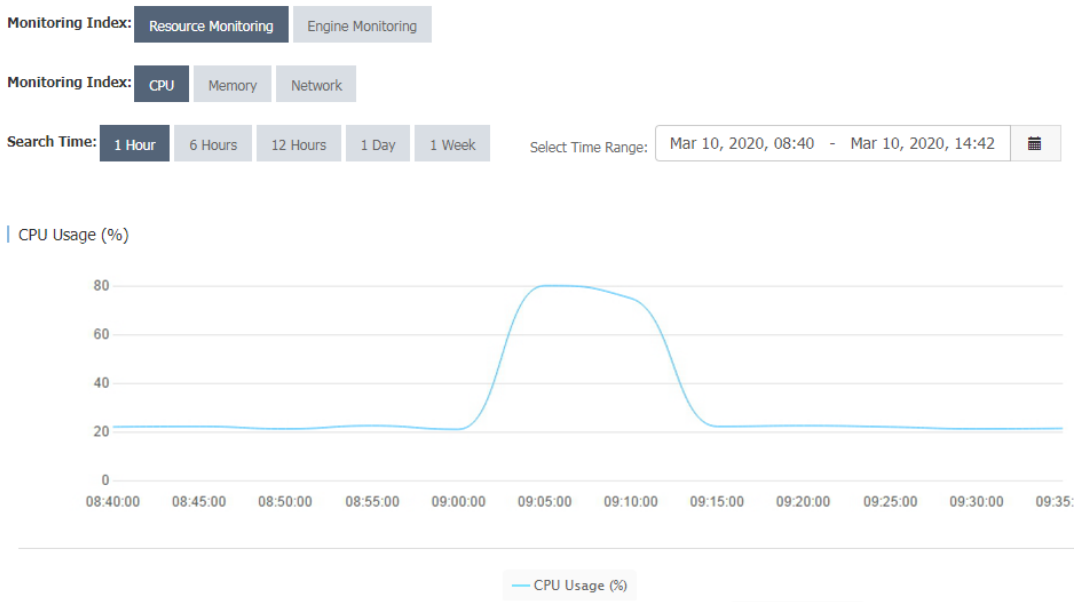
13.1.4.9.4.1. Example 1: PolarDB-X CPU utilization

Performance metrics change with the system business traffic.

The following describes the CPU utilization in two common cases:

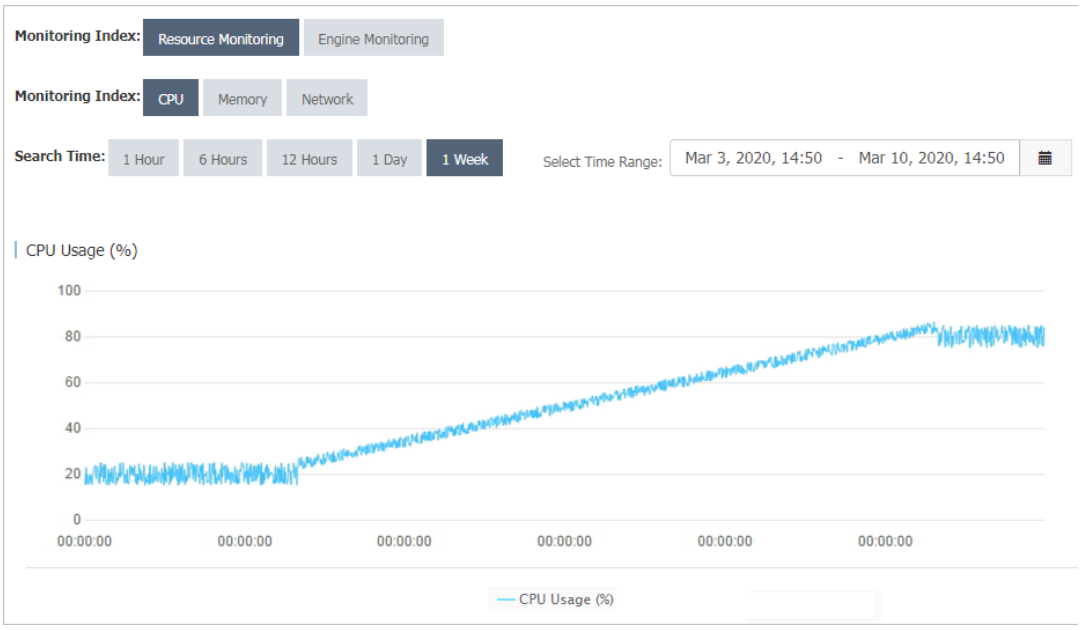
- An application has a shopping spree activity at 09:00 every morning. Therefore, the traffic of the system increases significantly at this time point. According to the monitoring data, the CPU utilization of the instance increased from 20% to about 80% from about 09:00, with the traffic peak lasting about 10 minutes.

  CPU utilization-1



- The system traffic keeps increasing with an application until it reaches a plateau. The monitored CPU utilization of the instance also reflects this change.

  CPU utilization-2



When the load on the instance changes with the business, you must pay close attention to the changes in metrics. If the CPU utilization exceeds the threshold, you must upgrade the specifications to alleviate the performance pressure.

You can set alert rules for instances in the console. When the average CPU utilization exceeds the preset threshold, the system sends short messages to the corresponding contacts. You can set the CPU utilization threshold as needed. We recommend that you set it to 80%.

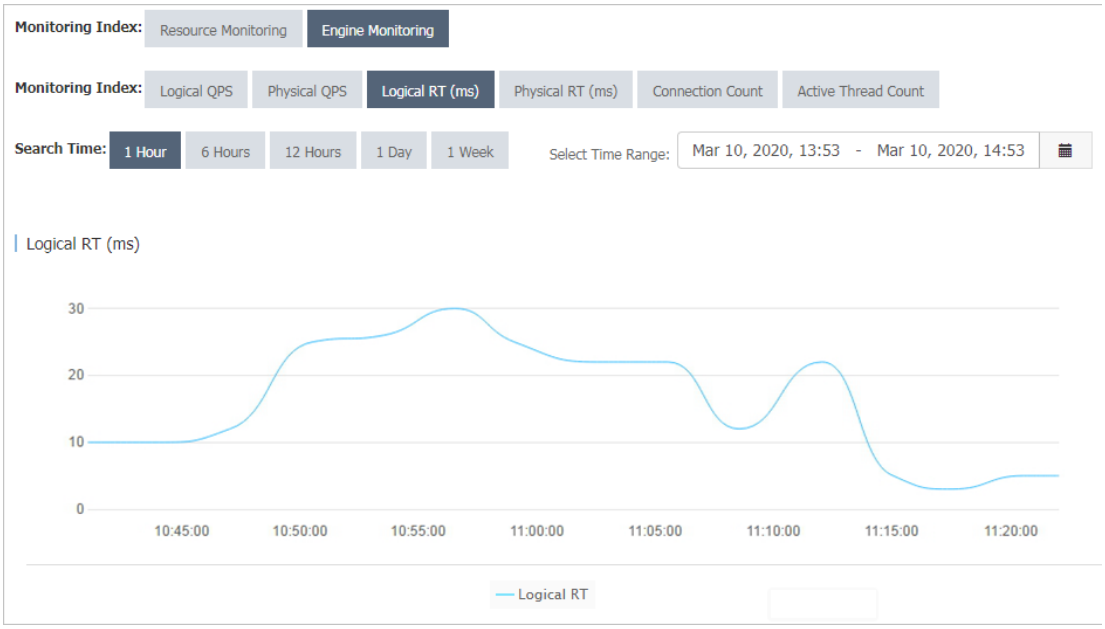
13.1.4.9.4.2. Example 2: Logical RT and physical RT

You can observe the difference between the logical response time (RT) and physical RT.

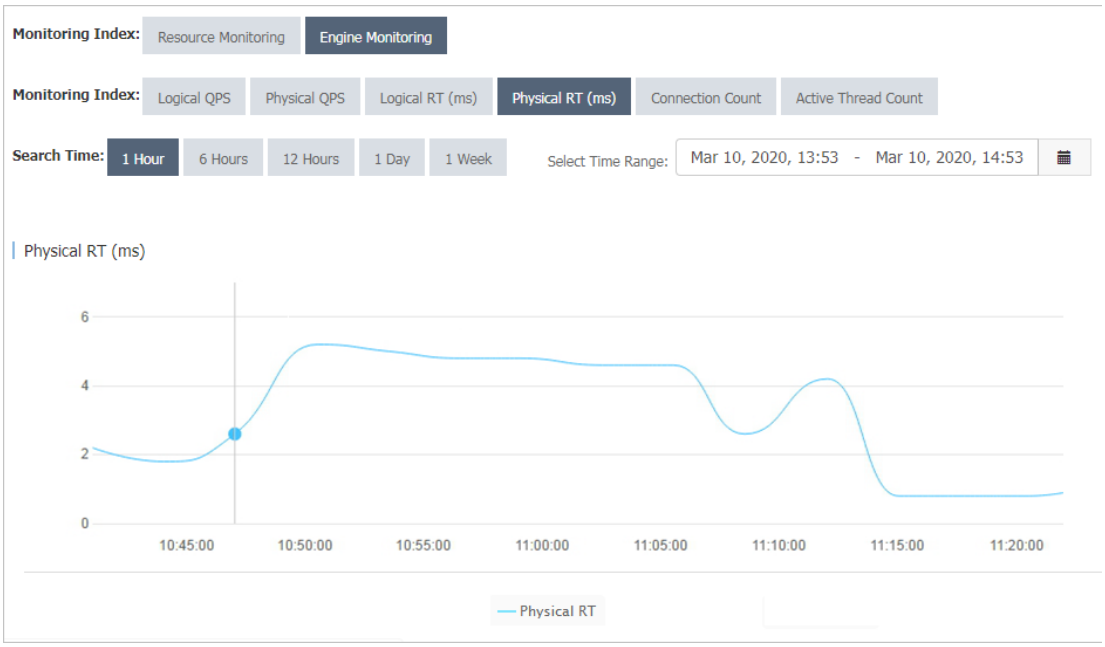
Logical RT refers to the time from when a instance receives a logical SQL statement to when it returns data to an application. Physical RT refers to the time from when a instance sends a physical SQL statement to an ApsaraDB RDS for MySQL instance to when it receives the data returned by the ApsaraDB RDS for MySQL instance.

If a logical SQL statement is partitioned into one or more physical SQL statements, the logical RT is greater than or equal to the physical RT. Ideally, performs only a few operations on the data returned by ApsaraDB RDS for MySQL. Therefore, logical RT is slightly longer than physical RT. Under special circumstances, physical SQL queries are run fast, while logical SQL queries take a long time to run. In this case, the logical RT and physical RT are as follows.

Logical RT



Physical RT



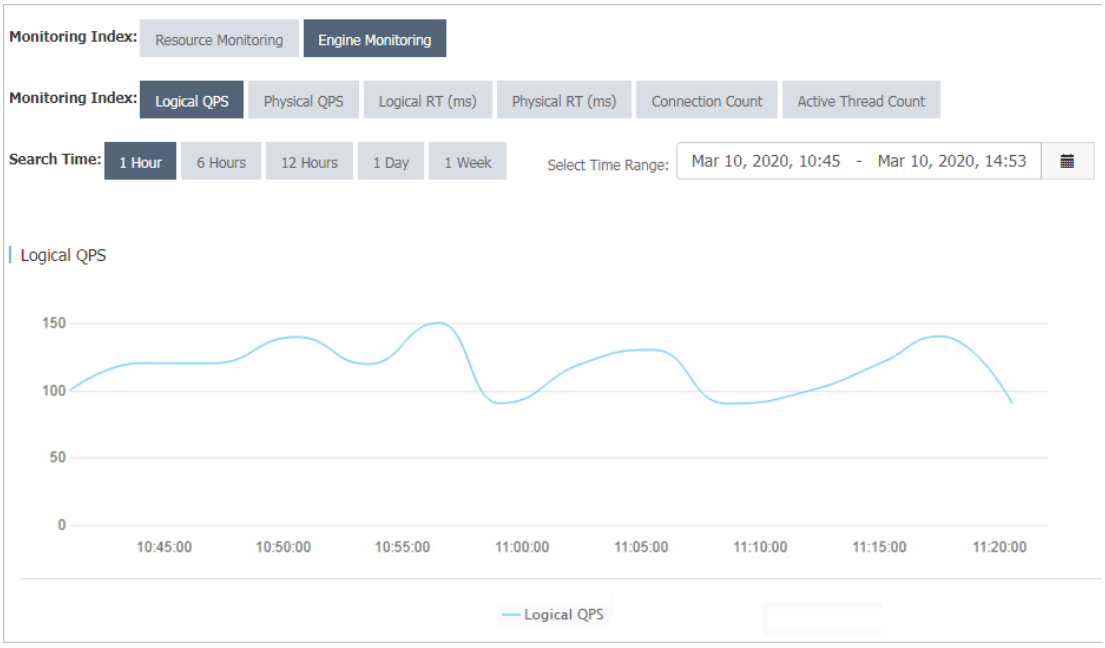
As shown in the preceding figures, the change trends of logical RT and physical RT in the two monitoring charts are basically the same, while logical RT fluctuates between 10 ms and 20 ms and physical RT fluctuates between 2 ms and 5 ms. This means that has a heavy load, which can be solved by upgrading the configuration. If both the logical RT and physical RT are high, you can upgrade the ApsaraDB RDS for MySQL configuration or optimize SQL statements on the ApsaraDB RDS for MySQL instance.

13.1.4.9.4.3. Example 3: Logical QPS and physical QPS

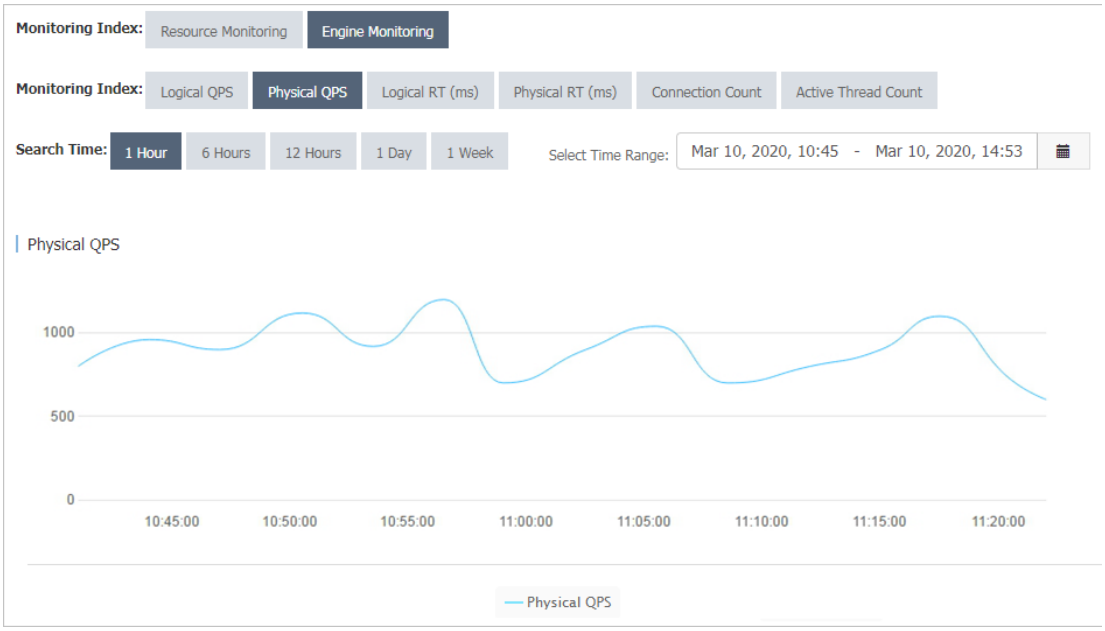
You can observe the difference between the logical queries per second (QPS) and physical QPS.

According to the monitoring data, the logical QPS and physical QPS have the same trends, but the difference between the two is relatively large and in a certain proportion.

Logical QPS



Physical QPS



As shown in the preceding figures, logical QPS fluctuates between 80 and 150, and physical QPS fluctuates between 700 and 1,200.

The reason is that generates physical SQL statements based on logical SQL statements. The ratio of logical SQL statements to physical SQL statements is not necessarily 1:1. For example, a logical table is created by using the following statement:

```
CREATE TABLE drds_user
(id int,
name varchar(30))
dbpartition by hash(id);
```

When the query condition contains the database shard key, pushes the logical SQL statement down to the ApsaraDB RDS for MySQL instance for execution. According to the execution plan, the number of physical SQL statements is 1:

```
mysql> explain select name from drds_user where id = 1;
+---------------------------+------------------------------------------------------------------------
-+--------+
| GROUP_NAME                | SQL
| PARAMS |
+---------------------------+------------------------------------------------------------------------
-+--------+
|     SANGUAN_BSQT_0001_RDS | select `drds_user`.`name` from `drds_user` where (`drds_user`.`id` = 1)
| {}     |
+---------------------------+------------------------------------------------------------------------
-+--------+
```

When the query does not contain the database shard key, partitions the logical SQL statement into multiple physical SQL statements. The following execution plan shows that there are eight physical SQL statements:

```
mysql> explain select name from drds_user where name = 'LiLei';
+----------- ---------------+------------------------------------------------------------------------
---------+--------+
| GROUP_NAME                | SQL
| PARAMS |
+---------------------------+------------------------------------------------------------------------
---------+--------+
| SANGUAN_BSQT_0001_RDS     | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` =
'LiLei') | {}     |
| SANGUAN_BSQT_0001_RDS     | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` =
'LiLei') | {}     |
| SANGUAN_BSQT_0001_RDS     | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` =
'LiLei') | {}     |
| SANGUAN_BSQT_0001_RDS     | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` =
'LiLei') | {}     |
| SANGUAN_BSQT_0001_RDS     | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` =
'LiLei') | {}     |
| SANGUAN_BSQT_0001_RDS     | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` =
'LiLei') | {}     |
| SANGUAN_BSQT_0001_RDS     | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` =
'LiLei') | {}     |
| SANGUAN_BSQT_0001_RDS     | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` =
'LiLei') | {}     |
+---------------------------+------------------------------------------------------------------------
---------+--------+
8 rows in set (0.06 sec)
```

Logical or physical QPS indicates the total number of logical or physical SQL statements processed per unit of time. When most SQL statements in the system contain the shard key, the ratio of logical QPS to physical QPS is close to 1:1. If the difference between the logical and physical QPS is too large, many SQL statements of the current application do not contain the shard key. In this case, check the SQL statements of the application to improve performance.

13.1.4.9.4.4. Example 4: High memory usage

The overly high memory usage of the instance is mostly caused by the large number of SQL queries in your application and the overlarge result set that is returned. If the memory usage of your instance remains at about 100%, perform the Restart a PolarDB-X instance operations to locate and optimize the slow SQL queries of your application.

13.1.4.10. View the instance version

You can view the instance version in two ways.

View the instance version in the console

1.
2.
3.
4. In the **Configuration Information** section, view the value of **Current Version**.

View the instance version by using the *version()* function

Connect to the instance through the MySQL command line and run the `SELECT version()` statement to view the version of the instance.

```
mysql> select version();
+----------------------------+
| VERSION()                  |
+----------------------------+
| 5.6.29-TDDL-5.1.28-1320920  |
+----------------------------+
1 row in set (0.00 sec)
```

In the preceding statement, 5.1.28-1320920 indicates the version of the instance.

13.1.5. Account management

13.1.5.1. Terms

This topic describes the terms of the account and permission system.

The usage of the account and permission system in is the same as that in MySQL. PolarDB-X supports statements such as `GRANT`, `REVOKE`, `SHOW GRANTS`, `CREATE USER`, `DROP USER`, and `SET PASSWORD`. Currently, PolarDB-X allows you to grant permissions at the database and table levels, but does not allow you to grant permissions at the global or column level.

For more information about the MySQL account and permission system, see MySQL official documentation.

Notice Accounts created by using `CREATE USER` in only exist in the instance and will not be synchronized to the backend ApsaraDB RDS for MySQL instances.

Account

An account is specified by the user name and hostname in the username@'host' format. Accounts with the same user name but different hostnames are different accounts. For example, lily@30.9.73.96 and lily@30.9.73.100 are two different accounts, and their passwords and permissions may be different.

After a database is created in the console, the system automatically creates two system accounts for the database: administrator account and read-only account. These two accounts are built-in accounts. You cannot delete them or modify their permissions.

- The administrator account name is the same as the database name. For example, if the database name is easydb, the administrator account name is easydb.
- The read-only account name is the database name suffixed with _RO. For example, if the database name is easydb, the read-only account name is easydb_RO.

Assume that the dreamdb and andordb databases are available. According to the preceding rules, the dreamdb database contains the dreamdb administrator account and the dreamdb_RO read-only account, while the andordb database contains the andordb administrator account and the andordb_RO read-only account.

Account rules

- Each administrator account has all permissions.
- Only the administrator account can create accounts and grant permissions. Other accounts can only be created and granted permissions by the administrator account.
- The administrator account is bound to a database and does not have permissions on other databases. It can only access the bound database, and cannot grant permissions of other databases to an account. For example, the easydb administrator account can only connect to the easydb database, and can only grant permissions of the easydb database or tables in the easydb database to an account.
- A read-only account has only the SELECT permission.

User name rules

- User names are case-insensitive.
- A user name must be 4 to 20 characters in length.
- A user name must start with a letter.
- A user name can contain letters and digits.

Password rules

- A password must be 6 to 20 characters in length.
- A password can contain letters, digits, and special characters (@ # $ % ^ & + =).

Hostname matching rules

- A hostname must be an IP address. It can contain underscores (_) and wildcards (%). A underscore (_) indicates a character and a wildcard (%) indicates no or more characters. Quote hostnames that contain wildcards with single quotations marks ('), for example, `lily@'30.9. %.%'` and `david@'%'`.
- If there are two user names that can be used to log on to the system, the user name with the longest prefix (the longest IP segment excluding wildcards) prevails. For example, if the `david@'30.9.12_.234'` and `david@'30.9.1%.234'` user names are available in the system, use `david@'30.9.12_.234'` to log on from the 30.9.127.234 host as david.
- When you enable the Virtual Private Cloud (VPC) access feature for a host, the IP address of the host changes. To avoid invalid configurations in the account and permission system, set the hostname to `'%'` to match any IP address.

Permission support

Permission support by level

- Global permission (not supported currently)
- Database-level permission (supported)
- Table-level permission (supported)
- Column-level permission (not supported currently)
- Subprogram-level permission (not supported currently)

Permissions

Currently, eight table-associated basic permissions are supported: `CREATE`, `DROP`, `ALTER`, `INDEX`, `INSERT`, `DELETE`, `UPDATE`, and `SELECT`.

- The `TRUNCATE` statement requires the table-level DROP permission.
- The `REPLACE` statement requires the table-level INSERT and DELETE permissions.
- `CREATE INDEX` and `DROP INDEX` statements are supported.
- The `CREATE SEQUENCE` statement requires the database-level CREATE permission.
- The `DROP SEQUENCE` statement requires the database-level DROP permission.
- The `ALTER SEQUENCE` statement requires the database-level ALTER permission.
- The `INSERT ON DUPLICATE UPDATE` statement requires the table-level INSERT and UPDATE permissions.

Permission rules

- Permissions are bound to an account (`username@'host'`) rather than a user name (`username`).
- An error occurs if the table does not exist during authorization.
- The database permissions in descending order are as follows: global permissions (not supported currently), database-level permissions, table-level permissions, and column-level permissions (not supported currently). A granted higher-level permission overwrites a lower-level permission. If you remove the higher-level permission, the lower-level permission is also removed.
- USAGE authorization is not supported.

13.1.5.2. Create an account

This topic describes how to create a account in the console and by using SQL statements.

Create an account in the console

1.
2.
3.
4.
5. On the **Account Management** page, click **Create Account** in the upper-right corner.
6. Configure the following parameters.

Parameter	Description
Database Account	Enter the account name. The account name must meet the following requirements: ○ The name must be 4 to 20 characters in length. ○ The name must start with a letter and end with a letter or digit. ○ The name can contain letters, digits, and underscores (_).

Parameter	Description
New Password	Enter an account password. The password must meet the following requirements: ○ The password must be 8 to 32 characters in length. ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ Special characters include ! @#$%^&*()_+-=
Confirm New Password	Enter the password again.
Authorization Databases	You can grant permissions on one or multiple databases to the account. i. Select one or more databases in the left-side section, and click **Add** to add them to the right-side section. ii. In the right-side section, select **Read/Write**, **Read-only**, **DDL Only**, or **DML Only**. **Note** You can also grant permissions on multiple authorized databases by clicking **Set All to Read-only**, **Set All to DDL Only**, **Set All to DML Only**, or **Set All to Read/Write** in the upper-right corner of the right-side section. The button in the upper-right corner changes as you click it. For example, after you click **Set All to Read-only**, the button is changed to **Set All to DDL Only**.

Create an account in the command line

Syntax rules:

```
CREATE USER user_specification [, user_specification] ...
  user_specification: user [ auth_option ]
  auth_option: IDENTIFIED BY 'auth_string'
```

Examples:

Create an account with the user name lily and password 123456, which can be used to log on only from 30.xx.xx.96.

```
CREATE USER lily@30.xx.xx.96 IDENTIFIED BY '123456';
```

Create an account named david with no password, which can be used to log on from any host.

```
CREATE USER david@'%';
```

13.1.5.3. Reset the password

When using , you can reset the password of your database account in the console or by using the command line.

Note

- Accounts with ROOT permissions cannot be deleted or modified.
- For data security, we recommend that you change your password periodically.

Reset the password in the console

1.
2.
3.
4.
5. Find the target account, and click **Reset Password**.
6. In the **Reset Account Password** dialog box, set **New Password** and **Confirm New Password**.

   > **Note** The password must meet the following requirements:
   > - The password must be 8 to 32 characters in length.
   > - The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.
   > - Special characters include ! @#$%^&*()_+-=

7. After you confirm that the password is correct, click **OK**.

Reset the password in the command line

Syntax rules:

```
SET PASSWORD FOR user = password_option
password_option: {
    PASSWORD('auth_string')
}
```

Examples:

Change the password of the account lily@30.xx.xx.96 to 123456.

```
SET PASSWORD FOR lily@30.xx.xx.96 = PASSWORD('123456')
```

13.1.5.4. Modify account permissions

You can modify the account permissions of your instances at any time when using .

Precautions

- Privileged accounts cannot be modified.
- In the console, you can only grant data manipulation language (DML), data definition language (DDL), read-only, and read/write permissions to standard accounts. To grant more permissions, use the command line.

Modify account permissions in the console

1.
2.
3.
4.
5. Find the target account and click **Modify Permission**.
6. In the **Modify Permissions** dialog box, **grant** or **remove** the permissions on one or more databases to or from the account.

- Authorize a database:

  Select one or more databases in the left-side section, and click **Add** to add them to the right-side section.

- Remove an authorized database:

  Select one or more databases in the right-side section, and click **Remove** to remove them. The removed databases are displayed in the left-side section.

- Modify permissions of an authorized database:

  In the right-side section, select **Read/Write**, **Read-only**, **DDL Only** or **DML Only**.

  > **Note** You can also grant permissions on multiple authorized databases by clicking **Set All to Read-only**, **Set All to DDL Only**, **Set All to DML Only**, or **Set All to Read/Write** in the upper-right corner of the right-side section.
  >
  > The button in the upper-right corner changes as you click it. For example, after you click **Set All to Read-only**, the button is changed to **Set All to DDL Only**.

7. After the configuration is complete, click **OK**.

GRANT statements

Syntax rules:

```
GRANT
    priv_type[, priv_type] ...
    ON priv_level
    TO user_specification [, user_specification] ...
    [WITH GRANT OPTION]
priv_level: {
  | db_name.*
  | db_name.tbl_name
  | tbl_name
}
user_specification:
    user [ auth_option ]
auth_option: {
    IDENTIFIED BY 'auth_string'
}
```

> **Notice**
>
> - If the account in the GRANT statement does not exist and no IDENTIFIED BY information is provided, an error message indicating that the account does not exist is returned.
> - If the account specified in the GRANT statement does not exist but the IDENTIFIED BY information is provided, the account is created and granted with the specified permission.

For example, in the easydb database, create an account named david, which can be used to log on from any host and has all the permissions on easydb.

Method 1: Create an account and then grant permissions to the account.

```
CREATE USER david@'%' IDENTIFIED BY 'your#password';
GRANT ALL PRIVILEGES ON easydb.* to david@'%';
```

Method 2: Create an account and grant permissions to the account by using one statement.

```
GRANT ALL PRIVILEGES ON easydb.* to david@'%' IDENTIFIED BY 'your#password';
```

In the easydb database, create an account named hanson, which can be used to log on from any host and has all the permissions on the easydb.employees table.

```
GRANT ALL PRIVILEGES ON easydb.employees to hanson@'%' IDENTIFIED BY 'your#password';
```

In the easydb database, create an account named hanson, which can be used to log on only from 192.xx. xx.10 and has the INSERT and SELECT permissions on the easydb.emp table.

```
GRANT INSERT,SELECT ON easydb.emp to hanson@'192.xx.xx.10' IDENTIFIED BY 'your#password';
```

In the easydb database, create a read-only account named actro, which can be used to log on from any host.

```
GRANT SELECT ON easydb.* to actro@'%' IDENTIFIED BY 'your#password';
```

REVOKE statements

Syntax rules:

- Delete the permissions at a certain level from an account. The permission level is specified by priv_level.

```
REVOKE
    priv_type
      [, priv_type] ...
    ON priv_level
    FROM user [, user] ...
```

- Delete all permissions of the account at the database and table levels.

```
REVOKE ALL PRIVILEGES, GRANT OPTION
    FROM user [, user] ...
```

Examples:

Delete the CREATE, DROP, and INDEX permissions from hanson@'%' on the easydb.emp table.

```
REVOKE CREATE,DROP,INDEX ON easydb.emp FROM hanson@'%';
```

Delete all permissions from the account lily@30.xx.xx.96.

```
REVOKE ALL PRIVILEGES,GRANT OPTION FROM lily@30.xx.xx.96;
```

Notice GRANT OPTION must be added to the statement for compatibility with MySQL.

SHOW GRANTS statements

Syntax rules:

```
SHOW GRANTS[FOR user@host];
```

Query all permissions:

```
SHOW GRANTS;
```

Query the permissions of an account:

```
SHOW GRANTS FOR user@host;
```

13.1.5.5. Delete an account

You can delete an account in the Cloud Native Distributed Database PolarDB-X (PolarDB-X) console or by using the command line.

Delete an account in the console

> **Note** You can only delete standard accounts that are created in the console.

1.
2.
3.
4.
5. Find the target account and click **Delete**.
6. In the **Delete Account** dialog box, click **OK**.

Delete an account by using the command line

Syntax rules:

```
DROP USER user [, user] ...
```

Examples:

Delete the account lily@30.xx.xx.96:

```
DROP USER lily@30.xx.xx.96;
```

13.1.6. Database management

13.1.6.1. Create a database

After you create a instance, create a database that runs on one or more ApsaraDB RDS for MySQL instances.

Prerequisites

- An ApsaraDB RDS for MySQL instance is created in the same department of the instance.
- The permissions on Resource Access Management (RAM) resources are granted. For more information about how to grant permissions, see *RAM management* in *Apsara Uni-manager user guide*.

1. Log on to the PolarDB-X console.
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More** > **Manage** from the Actions column of the instance to access the **Basic Information** page.
4. On the **Basic Information** page, click **Create Database** in the upper-right corner.
5. In the **Enter Basic Information** step, configure the following parameters:

Parameter	Description
Partition Mode	Select **Horizontal Partition**.
Database Name	Enter a custom name for the database. The name must meet the following requirements: ○ The name must be 2 to 24 characters in length. ○ The name must start with a letter and end with a letter or a digit. ○ The name can contain lowercase letters, digits, and underscores (_). ○ The name must be unique on the instance.
Character Set	Select utf8, gbk, latin1, or utf8mb4.
DRDS Link Password	Set the password for the database in . The password must meet the following requirements: ○ The password must be 8 to 30 characters in length. ○ The password must contain at least three of the following types: uppercase letters, lowercase letters, digits, and underscores (_).
Confirm Password	Enter the password again.

6. Click **Next**.
7. In the **Select RDS Instance** step, select **Buy New RDS Instance** or **Use Existing RDS Instance**.
   i. **Buy New RDS Instance**: Click the **Buy New RDS Instance** tab.
   ii. Set **Storage Type**, **Series**, **Instance Specifications**, **Storage Capacity**, **Availability Zone**, and **Quantity**.
   iii. Click **Next**.
   i. **Use Existing RDS Instance**: Click the **Use Existing RDS Instance** tab.
   ii. In the left-side section, click the ApsaraDB RDS for MySQL instances that you want to add.
   iii. Click > to add the selected instances to the **Selected RDS Instances** section on the right side.
   iv. Click **Next**.
8. After the precheck is successful in the **Precheck** step, click **Next**.

   > **Note** If the precheck fails, fix the issue based on the instructions on the page.

9. In the **Preview** step, click **Next**. Wait until the database is created.

13.1.6.2. View a database

After the database is created, you can view the basic information of the database in database management in the console.

Procedure

1.
2.
3.
4.
5. Find the target database, and click **Manage**. The **Basic Information** page of the database appears.

> **Note** On the **Basic Information** page, you can delete the database or reset the password.

What's next

is fully compatible with the MySQL protocol. You can use **Command Line URL** on the MySQL client to connect to the instance and enter the user name and password to log on to the database. When using the MySQL client, note the following points:

> **Note**
> - Some MySQL clients of earlier versions have limits on the user name length, which cannot be more than 16 characters. The database name and user name are the same. If the database name exceeds 16 characters, an error is reported.
> - When using the MySQL client, you must add the `-c` parameter to the hint command. In , HINT is implemented by using annotations. If the `-c` parameter is not added, the annotation is lost and the hint is lost.

13.1.6.3. Perform smooth scale-out

When the underlying storage of the logical database reaches the physical bottleneck, for example, when the remaining disk space is about 30%, you can smoothly scale it out to improve the performance. The smooth scale-out process is divided into four steps: configuration > migration > switchover > cleanup.

Configuration

> **Note** In smooth scale-out, ApsaraDB RDS for MySQL instances are added, and some source database shards are migrated to the new ApsaraDB RDS for MySQL instances. In this way, the overall data storage capacity is increased and the number of requests that a single ApsaraDB RDS for MySQL instance needs to process is reduced.

1.
2.
3.
4.
5. Find the target database, and click **Manage**. The **Basic Information** page of the database appears.
6. In the left-side navigation pane, choose **Configuration and Management > Scale-out Management**.
7. In the upper-right corner of the **Scale-out Management** page, click **Scale Out**.
8. Select **Smooth Scale-out**, and then click **Next**.
9. After all prechecks are passed on the **Precheck** page, click **Next**.

   > **Note** If a precheck fails, rectify the configuration as prompted.

10. On the **Select RDS Instance** page, you can click the **Buy New RDS Instance** or **Use Existing RDS Instance** tab.
    i. **Buy New RDS Instance**: Click the **Buy New RDS Instance** tab.
    ii. Set **Storage Type**, **Edition**, and **Storage Capacity**.
    iii. Click **Next**.
    i. **Use Existing RDS Instance**: Click the **Use Existing RDS Instance** tab.
    ii. Click the ApsaraDB RDS for MySQL instances to be added on the left.

iii. Click > to move the selected instances to the **Selected RDS Instances** section on the right.

iv. Click **Next**.

11. On the **Preview** page, click **Start Scale-out**.

> **Note** By default, the console evenly distributes the physical database shards to the ApsaraDB RDS for MySQL instances you added. You can also manually add or delete physical database shards to or from the new ApsaraDB RDS for MySQL instances.

12. Click the icon in the upper-right corner to view the details of the scale-out task.

Migration

Some physical database shards are migrated during smooth scale-out.

The migration does not change the data in the source database, and therefore it does not affect online services. Before the switchover, you can cancel the smooth scale-out operation through a rollback.

> **Note**
> - This is because before the switchover, the current scale-out operation does not have a real impact on the data in the source database.
> - During scale-out, the binary log files of the source ApsaraDB RDS for MySQL instance are not cleaned, which may result in insufficient disk space. Therefore, you must reserve sufficient disk space on the source ApsaraDB RDS for MySQL instance. Generally, the remaining disk space should be more than 30%. If the disk space cannot be guaranteed, you can submit a ticket to expand the ApsaraDB RDS for MySQL storage space.
> - To reduce the pressure of read operations on the source ApsaraDB RDS for MySQL instance, perform scale-out when the load on the source ApsaraDB RDS for MySQL instance is low.
> - During the scale-out, do not submit data description language (DDL) tasks in the console or connect to the instance to directly run DDL statements. Otherwise, the scale-out task may fail.
> - Make sure that all tables in the source database have primary keys before scale-out.

After historical data and incremental data are migrated, the migration progress reaches 100%. Then, you can **switch** the read and write traffic to the new ApsaraDB RDS for MySQL instance or **roll back** to cancel this scale-out.

Switchover

The switchover task switches the read and write traffic to the new ApsaraDB RDS for MySQL instance. The whole process takes 3 to 5 minutes. During the switchover process, the service is not affected except for one or two transient disconnections. Perform the switchover during off-peak hours.

1. In the upper-right corner of the **Basic Information** page, click the icon. The **Task Progress** dialog box appears.
2. In the **Task Progress** dialog box, click **Switch Over** and then **OK**.

   During the switchover process, a switchover task is generated and displayed in the Task Progress dialog box.
3. After the switchover is complete, the **Clean Up** button appears in the **Task Progress** dialog box, which means that the switchover task is complete.

Cleanup

In this step, the migrated database shards are deleted from the source ApsaraDB RDS for MySQL instance.

1. After switchover is complete, click **Clean Up** next to the target task.
2. Click **OK**. A cleanup task appears in the Task Progress dialog box.

> Note
> - The cleanup task is an asynchronous task. You can view the execution status in the Task Progress dialog box.
> - After the cleanup task is complete, the smooth scale-out process ends. The new ApsaraDB RDS for MySQL instance becomes the storage node of the logical database.
> - Currently, you can implement smooth scale-out by migrating physical database shards. If no further scale-out is allowed after the number of database shards exceeds the capacity of a single ApsaraDB RDS for MySQL instance, you can submit a ticket to apply for increasing the number of database shards and scaling out the database. In this case, Hash calculation is performed again to reallocate data.
> - The cleanup task deletes database shards that are no longer used after the current scale-out. You can back up the database shards before running the cleanup task.
> - The cleanup operation brings pressure to databases. We recommend that you perform this operation during off-peak hours.

13.1.6.4. View database monitoring information

displays the historical monitoring information of a database in two dimensions: data metrics and query time.

Procedure

1.
2.
3.
4. In the left-side navigation pane, choose **Monitoring and Alerts > Database Monitoring**.
5. Select **Data Indexes** and **Query Time**. Then, the corresponding monitoring information appears.

> Note For more information about instance-level monitoring, see View monitoring information.

13.1.6.5. Set the IP address whitelist

provides the access control function. Only IP addresses in the whitelist of a database can access the database.

Procedure

1.
2.
3.
4.
5.
6. On the **Basic Information** page of the database, choose **Data Security > Whitelist Settings** from the left-side navigation pane.
7. On the Whitelist Settings page, click **Manually Modify**.
8. Enter the IP address that is allowed to access the database, and click **OK**.

> **Note**
> - The whitelist supports the following formats:
>   - IP address, for example, 192.168.1.1.
>   - CIDR IP address, for example, 192.168.1.1/24.
>   - IP address with an asterisk (*) as the wildcard, for example, 192.168.1. *, indicating that access is allowed from any host with an IP address in the range of 192.168.1.1 to 192.168.1.254.
>   - CIDR block, for example, 192.168.1.1-192.168.1.254.
> - If you want to add multiple IP addresses or CIDR blocks, separate them with commas (,) without spaces, for example, 192.168.0.1,172.16.213.9.

13.1.6.6. Delete a database

This topic describes how to delete a database in the Cloud Native Distributed Database PolarDB-X (PolarDB-X) console.

Procedure

1.
2.
3.
4.
5. Find the target database and click **Delete**.

   > **Warning** You cannot recover databases that have been deleted. Exercise caution when you perform this operation.

6. In the **Delete Database** dialog box, click **OK**.

13.1.6.7. Fix database shard connections

Context

When using a instance, you need to access ApsaraDB RDS for MySQL. If the network configuration of the connected ApsaraDB RDS for MySQL instance changes, for example, if the zone is switched or the network type is changed from classic to Virtual Private Cloud (VPC), the network connection between the instance and the ApsaraDB RDS for MySQL instance is broken. As a result, the instance cannot access the ApsaraDB RDS for MySQL instance. In this case, you must manually fix the database shard link in the console to restore the network connection from the instance to the ApsaraDB RDS for MySQL instance.

Procedure

1.
2.
3.
4.
5.
6. In the **Shortcuts** section, click **Fix Database Shard Connections**.
7. In the pop-up window, click **OK**.

13.1.7. Custom control commands

provides a series of auxiliary SQL commands to help you conveniently use .

13.1.7.1. Overview

provides unique auxiliary statements for you to use and maintain .

Syntax description: The identifier provided by the user is in [] and optional content is in (). In addition, this document applies to the current version. If some statements are unavailable, the version is earlier than required.

13.1.7.2. Help statements

This topic describes all the auxiliary SQL commands of and their descriptions.

SHOW HELP statements:

```
mysql> show help;
+----------------------------------------+-------------------------------------------------------+
-----------------------------------------+
| STATEMENT                              | DESCRIPTION                                           |
EXAMPLE                                  |
+----------------------------------------+-------------------------------------------------------+
-----------------------------------------+
| show rule                              | Report all table rule                                 |
|
| show rule from TABLE                   | Report table rule                                     |
show rule from user                      |
| show full rule from TABLE              | Report table full rule                                |
show full rule from user                 |
| show topology from TABLE               | Report table physical topology                        |
show topology from user                  |
| show partitions from TABLE             | Report table dbPartition or tbPartition columns       |
show partitions from user                |
| show broadcasts                        | Report all broadcast tables                           |
|
| show datasources                       | Report all partition db threadPool info               |
|
| show node                              | Report master/slave read status                       |
|
| show slow                              | Report top 100 slow sql                               |
|
| show physical_slow                     | Report top 100 physical slow sql                      |
|
| clear slow                             | Clear slow data                                       |
|
| trace SQL                              | Start trace sql, use show trace to print profiling data |
trace select count(*) from user; show trace |
| show trace                             | Report sql execute profiling info                     |
|
| explain SQL                            | Report sql plan info                                  |
explain select count(*) from user        |
| explain detail SQL                     | Report sql detail plan info                           |
explain detail select count(*) from user |
| explain execute SQL                    | Report sql on physical db plan info                   |
explain execute select count(*) from user |
| show sequences                         | Report all sequences status                           |
|
| create sequence NAME [start with COUNT] | Create sequence                                       |
create sequence test start with 0        |
| alter sequence NAME [start with COUNT]  | Alter sequence                                        |
alter sequence test start with 100000    |
| drop sequence NAME                     | Drop sequence                                         |
drop sequence test                       |
+----------------------------------------+-------------------------------------------------------+
-----------------------------------------+
20 rows in set (0.00 sec)
```

13.1.7.3. Statements for viewing rules and node topologies

SHOW RULE [FROM tablename]

Usage notes:

- `show rule`: shows the partitioning status of each logical table in a database.
- `show rule from tablename`: shows the partitioning status of a specified logical table in a database.

The following describes the meanings of important columns:

- BROADCAST: indicates whether the table is a broadcast table. 0 indicates "No" and 1 indicates "Yes".
- DB_PARTITION_KEY: indicates the database shard key. If no database shards exist, the parameter value is NULL.
- DB_PARTITION_POLICY: indicates the database sharding policy. Options are Hash and date policies such as YYYYMM, YYYYDD, and YYYYWEEK.
- DB_PARTITION_COUNT: indicates the number of database shards.
- TB_PARTITION_KEY: indicates the table shard key. If no table shards exist, the parameter value is NULL.
- TB_PARTITION_POLICY: indicates the table sharding policy. Options are Hash and date policies such as MM, DD, MMDD, and WEEK.
- TB_PARTITION_COUNT: indicates the number of table shards.

```
mysql> show rule;
+------+--------------+-----------+------------------+---------------------+--------------------+----
--------------+---------------------+--------------------+
| ID   | TABLE_NAME   | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT | TB_
PARTITION_KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+------+--------------+-----------+------------------+---------------------+--------------------+----
--------------+---------------------+--------------------+
|    0 | dept_manager |         0 |                  | NULL                | 1                  |
| NULL                | 1                   |
|    1 | emp          |         0 | emp_no           | hash                | 8                  | id
| hash                | 2                   |
|    2 | example      |         0 | shard_key        | hash                | 8                  |
| NULL                | 1                   |
+------+--------------+-----------+------------------+---------------------+--------------------+----
--------------+---------------------+--------------------+
3 rows in set (0.01 sec)
```

SHOW FULL RULE [FROM tablename]

You can run this SQL statement to view the sharding rules of logical tables in a database. It displays more detailed information than the SHOW RULE command.

The following describes the meanings of important columns:

- BROADCAST: indicates whether the table is a broadcast table. 0 indicates "No" and 1 indicates "Yes".
- JOIN_GROUP: a reserved field.
- ALLOW_FULL_TABLE_SCAN: indicates whether to allow data querying when no table shard key is specified for database or table sharding. If this parameter is set to True, each physical table is scanned to find data that meets the condition, which is a full table scan.
- DB_NAME_PATTERN: The value 0 between {} in DB_NAME_PATTERN is a placeholder. When the SQL statement is run, this value is replaced by the value of DB_RULES_STR, with the number of digits unchanged. For example, if the value of DB_NAME_PATTERN is SEQ_{0000}_RDS and the value of DB_RULES_STR is [1,2,3,4], four DB_NAME values are generated: SEQ_0001_RDS, SEQ_0002_RDS, SEQ_0003_RDS, and SEQ_0004_RDS.
- DB_RULES_STR: indicates the database sharding rule.
- TB_NAME_PATTERN: The value 0 between {} in TB_NAME_PATTERN is a placeholder. When the SQL statement is run, this value is replaced by the value of TB_RULES_STR, with the number of digits unchanged. For example, if the value of TB_NAME_PATTERN is table_{00} and the value of TB_RULES_STR is [1,2,3,4,5,6,7,8], eight tables are generated: table_01, table_02, table_03, table_04, table_05, table_06, table_07, and table_08.
- TB_RULES_STR: indicates the table sharding rule.
- PARTITION_KEYS: indicates a set of database and table shard keys. When database sharding and table sharding

coexist, the database shard key is placed before the table shard key.

- DEFAULT_DB_INDEX: indicates the database shard in which a single database and a single table are stored.

```
mysql> show full rule;
+------+--------------+-----------+------------+----------------------+-----------------------------
----------------------+-----------------------------------------------------------+-----------------+-----
------------------------------+----------------+-----------------------------------------------------+
| ID   | TABLE_NAME   | BROADCAST | JOIN_GROUP | ALLOW_FULL_TABLE_SCAN | DB_NAME_PATTERN
| DB_RULES_STR                                        | TB_NAME_PATTERN | TB_RULES_STR
| PARTITION_KEYS | DEFAULT_DB_INDEX                                |
+------+--------------+-----------+------------+----------------------+-----------------------------
----------------------+-----------------------------------------------------------+-----------------+-----
------------------------------+----------------+-----------------------------------------------------+
|    0 | dept_manager |         0 | NULL       |                    0 | SEQ_TEST_1487767780814RGKKSE
Q_TEST_WNJG_0000_RDS   | NULL                                                |                 | dept_manager    | NULL
| NULL           | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0000_RDS |
|    1 | emp          |         0 | NULL       |                    1 | SEQ_TEST_1487767780814RGKKSE
Q_TEST_WNJG_{0000}_RDS | ((#emp_no,1,8#).longValue().abs() % 8)              | emp_{0}         | ((#i
d,1,2#).longValue().abs() % 2) | emp_no      id | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0000_RDS |
|    2 | example      |         0 | NULL       |                    1 | SEQ_TEST_1487767780814RGKKSE
Q_TEST_WNJG_{0000}_RDS | ((#shard_key,1,8#).longValue().abs() % 8).intdiv(1) | example         | NULL
| shard_key      | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0000_RDS |
+------+--------------+-----------+------------+----------------------+-----------------------------
----------------------+-----------------------------------------------------------+-----------------+-----
------------------------------+----------------+-----------------------------------------------------+
3 rows in set (0.01 sec)
```

SHOW TOPOLOGY FROM tablename

You can run this SQL statement to view the topology of a specified logical table, that is, the database shards to which data in the logical table is partitioned and the table shards in each database shard.

```
mysql> show topology from emp;
+------+-----------------------------------------------------+------------+
| ID   | GROUP_NAME                                          | TABLE_NAME |
+------+-----------------------------------------------------+------------+
|    0 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0000_RDS | emp_0      |
|    1 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0000_RDS | emp_1      |
|    2 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0001_RDS | emp_0      |
|    3 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0001_RDS | emp_1      |
|    4 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0002_RDS | emp_0      |
|    5 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0002_RDS | emp_1      |
|    6 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0003_RDS | emp_0      |
|    7 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0003_RDS | emp_1      |
|    8 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0004_RDS | emp_0      |
|    9 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0004_RDS | emp_1      |
|   10 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0005_RDS | emp_0      |
|   11 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0005_RDS | emp_1      |
|   12 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0006_RDS | emp_0      |
|   13 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0006_RDS | emp_1      |
|   14 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0007_RDS | emp_0      |
|   15 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0007_RDS | emp_1      |
+------+-----------------------------------------------------+------------+
16 rows in set (0.01 sec)
```

SHOW PARTITIONS FROM tablename

You can run this SQL statement to view the set of database shard keys and table shard keys, which are separated by commas (,). If the final result contains two values, both database sharding and table sharding are performed. The first value is the database shard key and the second value is the table shard key. If only one value is returned, only database sharding is performed. This value is the database shard key.

```
mysql> show partitions from emp;
+-----------+
| KEYS      |
+-----------+
| emp_no,id |
+-----------+
1 row in set (0.00 sec)
```

SHOW BROADCASTS

You can run this SQL statement to view the list of broadcast tables.

```
mysql> show broadcasts;
+------+------------+
| ID   | TABLE_NAME |
+------+------------+
|    0 | brd2       |
|    1 | brd_tbl    |
+------+------------+
2 rows in set (0.01 sec)
```

SHOW DATASOURCES

You can run this SQL statement to view the information about the underlying storage, including the database name, database group name, connection URL, user name, storage type, read/write weight, and connection pool information.

The following describes the meanings of important columns:

- SCHEMA: indicates the database name.
- GROUP: indicates the database group name. Grouping aims to manage multiple groups of databases that have identical data, such as the primary and secondary databases after data replication through ApsaraDB RDS for MySQL. It is mainly used for read/write splitting and primary/secondary switchovers.
- URL: indicates the connection information of the underlying ApsaraDB RDS for MySQL instance.
- TYPE: indicates the type of the underlying storage. Currently, only ApsaraDB RDS for MySQL instances are supported.
- READ_WEIGHT: indicates the read weight of the database. When the primary ApsaraDB RDS for MySQL instance is under a heavy load of many read requests, you can use the read/write splitting function of to distribute the read traffic. automatically identifies the read and write traffic. It directs the write traffic to the primary ApsaraDB RDS for MySQL instance and the read traffic to all ApsaraDB RDS for MySQL instances based on the configured weight.
- WRITE_WEIGHT: indicates the write weight. For more information, see READ_WEIGHT.

```
mysql> show datasources;
+------+----------------------------+-------------------------------------------------+---------------
--------------------------------+-------------------------------------------------------------------------
----------------+-----------+-------+------+------+------+-------------+----------+--------------+-
-------------+-------------------------------------------------+-------------+--------------+
| ID   | SCHEMA                     | NAME                                            | GROUP
| URL                                                                                                    | USER      | TYPE
| INIT | MIN  | MAX  | IDLE_TIMEOUT | MAX_WAIT | ACTIVE_COUNT | POOLING_COUNT | ATOM
| READ_WEIGHT | WRITE_WEIGHT |
+------+----------------------------+-------------------------------------------------+---------------
--------------------------------+-------------------------------------------------------------------------
----------------+-----------+-------+------+------+------+-------------+----------+--------------+-
-------------+-------------------------------------------------+-------------+--------------+
|    0 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0000_iiab_1 | SEQ_TEST_14877
67780814RGKKSEQ_TEST_WNJG_0000_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/se
q_test_wnjg_0000 | jnkinsea0 | mysql | 0    | 24   | 72   | 15           | 5000     | 0            |
1             | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0000_iiab | 10          | 10           |
|    1 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0001_iiab_2 | SEQ_TEST_14877
67780814RGKKSEQ_TEST_WNJG_0001_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/se
q_test_wnjg_0001 | jnkinsea0 | mysql | 0    | 24   | 72   | 15           | 5000     | 0            |
1             | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0001_iiab | 10          | 10           |
|    2 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0002_iiab_3 | SEQ_TEST_14877
67780814RGKKSEQ_TEST_WNJG_0002_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/se
q_test_wnjg_0002 | jnkinsea0 | mysql | 0    | 24   | 72   | 15           | 5000     | 0            |
1             | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0002_iiab | 10          | 10           |
|    3 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0003_iiab_4 | SEQ_TEST_14877
67780814RGKKSEQ_TEST_WNJG_0003_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/se
q_test_wnjg_0003 | jnkinsea0 | mysql | 0    | 24   | 72   | 15           | 5000     | 0            |
1             | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0003_iiab | 10          | 10           |
|    4 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0004_iiab_5 | SEQ_TEST_14877
67780814RGKKSEQ_TEST_WNJG_0004_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/se
q_test_wnjg_0004 | jnkinsea0 | mysql | 0    | 24   | 72   | 15           | 5000     | 0            |
1             | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0004_iiab | 10          | 10           |
|    5 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0005_iiab_6 | SEQ_TEST_14877
67780814RGKKSEQ_TEST_WNJG_0005_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/se
q_test_wnjg_0005 | jnkinsea0 | mysql | 0    | 24   | 72   | 15           | 5000     | 0            |
1             | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0005_iiab | 10          | 10           |
|    6 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0006_iiab_7 | SEQ_TEST_14877
67780814RGKKSEQ_TEST_WNJG_0006_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/se
q_test_wnjg_0006 | jnkinsea0 | mysql | 0    | 24   | 72   | 15           | 5000     | 0            |
1             | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0006_iiab | 10          | 10           |
|    7 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0007_iiab_8 | SEQ_TEST_14877
67780814RGKKSEQ_TEST_WNJG_0007_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/se
q_test_wnjg_0007 | jnkinsea0 | mysql | 0    | 24   | 72   | 15           | 5000     | 0            |
1             | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0007_iiab | 10          | 10           |
+------+----------------------------+-------------------------------------------------+---------------
--------------------------------+-------------------------------------------------------------------------
----------------+-----------+-------+------+------+------+-------------+----------+--------------+-
-------------+-------------------------------------------------+-------------+--------------+
8 rows in set (0.01 sec)
```

SHOW NODE

You can run this SQL statement to view the accumulative number of read and write operations and accumulative read/write weights of a physical database.

The following describes the meanings of important columns:

- MASTER_READ_COUNT: indicates the accumulative number of read-only queries processed by the primary

ApsaraDB RDS for MySQL instance.

- SLAVE_READ_COUNT: indicates the accumulative number of read-only queries processed by the secondary ApsaraDB RDS for MySQL instances.
- MASTER_READ_PERCENT: indicates the actual percentage of read-only queries processed by the primary ApsaraDB RDS for MySQL instance, not the configured percentage.
- SLAVE_READ_PERCENT: indicates the actual percentage of read-only queries processed by the secondary ApsaraDB RDS for MySQL instances, not the configured percentage.

> **Note**
> - Read-only queries in transactions are sent to the primary ApsaraDB RDS for MySQL instance.
> - The `MASTER_READ_PERCENT` and `SLAVE_READ_PERCENT` fields indicate the accumulative historical data. After the read/write weight ratio has been changed, these values do not immediately reflect the latest read/write weight ratio, which appears after a long period of time has passed.

```
mysql> show node;
+------+-----------------------------------------------------+------------------+------------------+---
------------------+--------------------+
| ID   | NAME                                                | MASTER_READ_COUNT | SLAVE_READ_COUNT | MA
STER_READ_PERCENT | SLAVE_READ_PERCENT |
+------+-----------------------------------------------------+------------------+------------------+---
------------------+--------------------+
| 0    | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0000_RDS |                12 |                0 | 10
0%                | 0%                 |
| 1    | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0001_RDS |                 0 |                0 | 0%
| 0%                |
| 2    | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0002_RDS |                 0 |                0 | 0%
| 0%                |
| 3    | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0003_RDS |                 0 |                0 | 0%
| 0%                |
| 4    | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0004_RDS |                 0 |                0 | 0%
| 0%                |
| 5    | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0005_RDS |                 0 |                0 | 0%
| 0%                |
| 6    | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0006_RDS |                 0 |                0 | 0%
| 0%                |
| 7    | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0007_RDS |                 0 |                0 | 0%
| 0%                |
+------+-----------------------------------------------------+------------------+------------------+---
------------------+--------------------+
8 rows in set (0.01 sec)
```

13.1.7.4. SQL tuning statements

SHOW [FULL] SLOW [WHERE expr] [limit expr]

SQL statements that take more than 1 second to execute are slow SQL statements. Slow logical SQL statements are the slow SQL statements sent from an application to a instance.

- `SHOW SLOW`: You can run this SQL statement to view the 100 slowest logical SQL queries that are recorded since the instance is started or the last time when `CLEAR SLOW` is executed.

  > **Note** The recorded 100 slowest SQL queries are stored in the system. When the PolarDB-X instance is restarted or executes `CLEAR SLOW`, these queries will be discarded.

- `SHOW FULL SLOW`: You can run this SQL statement to view all the slow logical SQL queries that are recorded and persisted to the built-in database of the instance since the PolarDB-X instance is started. The upper limit for the number of records is specified in the specifications of the PolarDB-X instance. The instance scrolls to delete the earliest slow SQL statements. If the specifications of the PolarDB-X instance is 4-core 4 GB, a maximum of 10,000 slow SQL statements can be recorded, including slow logical and physical SQL statements. If the specifications of the PolarDB-X instance is 8-core 8 GB, a maximum of 20,000 slow SQL statements can be recorded, including slow logical and physical SQL statements. The same rule applies to other specifications.

The following describes the meanings of important columns:

- HOST: the IP address of the host from which the SQL statement is sent.
- START_TIME: the time when the SQL statement starts to be executed.
- EXECUTE_TIME: the time when the SQL statement is executed.
- AFFECT_ROW: For data manipulation language (DML) statements, this parameter indicates the number of affected rows. For query statements, this parameter indicates the number of returned records.

```
mysql> show slow where execute_time > 1000 limit 1;
+-----------+---------------------+--------------+------------+-----------+
| HOST      | START_TIME          | EXECUTE_TIME | AFFECT_ROW | SQL       |
+-----------+---------------------+--------------+------------+-----------+
| 127.0.0.1 | 2016-03-16 13:02:57 |         2785 |          7 | show rule |
+-----------+---------------------+--------------+------------+-----------+
1 row in set (0.02 sec)
```

SHOW [FULL] PHYSICAL_SLOW [WHERE expr] [limit expr]

SQL statements that take more than 1 second to execute are slow SQL statements. Slow logical SQL statements are the slow SQL statements sent from an application to a instance.

- `SHOW SLOW`: You can run this SQL statement to view the 100 slowest logical SQL queries that are recorded since the instance is started or the last time when `CLEAR SLOW` is executed.

  > **Note** The recorded 100 slowest SQL queries are stored in the system. When the PolarDB-X instance is restarted or executes `CLEAR SLOW`, these queries will be discarded.

- `SHOW FULL SLOW`: You can run this SQL statement to view all the slow logical SQL queries that are recorded and persisted to the built-in database of the instance since the PolarDB-X instance is started. The upper limit for the number of records is specified in the specifications of the PolarDB-X instance. The instance scrolls to delete the earliest slow SQL statements. If the specifications of the PolarDB-X instance is 4-core 4 GB, a maximum of 10,000 slow SQL statements can be recorded, including slow logical and physical SQL statements. If the specifications of the PolarDB-X instance is 8-core 8 GB, a maximum of 20,000 slow SQL statements can be recorded, including slow logical and physical SQL statements. The same rule applies to other specifications.

The following describes the meanings of important columns:

- GROUP_NAME: the name of the group to which the database that executes the SQL statement belongs.
- START_TIME: the time when the SQL statement starts to be executed.
- EXECUTE_TIME: the time when the SQL statement is executed.
- AFFECT_ROW: For data manipulation language (DML) statements, this parameter indicates the number of affected rows. For query statements, this parameter indicates the number of returned records.

```
mysql> show physical_slow;
+----------------+-----------------------------------+--------------------+--------------+----------
--------+------------------------+----------------------+------------+-----------------+
| GROUP_NAME     | DBKEY_NAME                        | START_TIME         | EXECUTE_TIME | SQL_EXECU
TE_TIME | GETLOCK_CONNECTION_TIME | CREATE_CONNECTION_TIME | AFFECT_ROW | SQL             |
+----------------+-----------------------------------+--------------------+--------------+----------
--------+------------------------+----------------------+------------+-----------------+
| TDDL5_00_GROUP | db218249098_sqa_zmf_tddl5_00_3309 | 2016-03-16 13:05:38 |         1057 |
1011 |                      0 |                      0 |          1 | select sleep(1) |
+----------------+-----------------------------------+--------------------+--------------+----------
--------+------------------------+----------------------+------------+-----------------+
1 row in set (0.01 sec)
```

CLEAR SLOW

You can run this SQL statement to clear the 100 slowest logical SQL queries and the 100 slowest physical SQL queries that are recorded since the instance is started or the last time when `CLEAR SLOW` is executed.

> **Note** Both `SHOW SLOW` and `SHOW PHYSICAL_SLOW` can be executed to display the 100 slowest SQL statements. If `CLEAR SLOW` has not been executed for a long time, these SQL statements may have been recorded a long time ago. Therefore, after SQL tuning statements are executed, we recommend that you execute `CLEAR SLOW`. After the system runs for a while, check the tuning results of slow SQL statements.

```
mysql> clear slow;
Query OK, 0 rows affected (0.00 sec)
```

EXPLAIN SQL

You can run this SQL statement to view the execution plan of a specified SQL statement in the . Note that this SQL statement is not truly executed.

Example:

You can run this SQL statement to view the execution plan of the SQL `select * from doctest` statement. The doctest table is stored in database shards according to values in the id column. According to the execution plan, the SQL statement will be routed to each database shard for execution, and the execution results will be aggregated.

```
mysql> explain select * from doctest;
+-----------------------------------------------+-------------------------------------+---------+
| GROUP_NAME                                    | SQL                                 | PARAMS |
+-----------------------------------------------+-------------------------------------+---------+
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0000_RDS | select `doctest`.`id` from `doctest` | {}     |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0001_RDS | select `doctest`.`id` from `doctest` | {}     |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0002_RDS | select `doctest`.`id` from `doctest` | {}     |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0003_RDS | select `doctest`.`id` from `doctest` | {}     |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0004_RDS | select `doctest`.`id` from `doctest` | {}     |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0005_RDS | select `doctest`.`id` from `doctest` | {}     |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0006_RDS | select `doctest`.`id` from `doctest` | {}     |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0007_RDS | select `doctest`.`id` from `doctest` | {}     |
+-----------------------------------------------+-------------------------------------+---------+
8 rows in set (0.00 sec)
```

You can run this SQL statement to view the execution plan of the SQL `select * from doctest where id = 1` statement. The doctest table is stored in database shards according to values in the id column. According to the execution plan, the PolarDB-X instance will calculate a specified database shard based on the shard key, which is id, directly route the SQL statement to the database shard, and aggregate the execution results.

```
mysql> explain select * from doctest where id = 1;
+-------------------------------------------------+-------------------------------------------------
-------------+---------+
| GROUP_NAME                                      | SQL
| PARAMS |
+-------------------------------------------------+-------------------------------------------------
-------------+---------+
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0001_RDS | select `doctest`.`id` from `doctest` where (`docte
st`.`id` = 1) | {}     |
+-------------------------------------------------+-------------------------------------------------
-------------+---------+
1 row in set (0.01 sec)
```

EXPLAIN DETAIL SQL

You can run this SQL statement to view the execution plan of a specified SQL statement in the . Note that this SQL statement is not truly executed.

```
mysql> explain detail select * from doctest where id = 1;
+-------------------------------------------------+-------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------+---------+
| GROUP_NAME                                      | SQL
| PARAMS |
+-------------------------------------------------+-------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------+---------+
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0001_RDS | Query from doctest as doctest
    keyFilter:doctest.id = 1
    queryConcurrency:SEQUENTIAL
    columns:[doctest.id]
    tableName:doctest
    executeOn:DOCTEST_1488704345426RCUPDOCTEST_CAET_0001_RDS
 | NULL   |
+-------------------------------------------------+-------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------+---------+
1 row in set (0.02 sec)
```

EXPLAIN EXECUTE SQL

You can run this SQL statement to view the execution plan of underlying storage. This statement is equivalent to the MySQL EXPLAIN statement.

```
mysql> explain execute select * from tddl_mgr_log limit 1;
+----+-------------+--------------+------+---------------+------+---------+------+------+-------+
| id | select_type | table        | type | possible_keys | key  | key_len | ref  | rows | Extra |
+----+-------------+--------------+------+---------------+------+---------+------+------+-------+
|  1 | SIMPLE      | tddl_mgr_log | ALL  | NULL          | NULL | NULL    | NULL |    1 | NULL  |
+----+-------------+--------------+------+---------------+------+---------+------+------+-------+
1 row in set (0.07 sec)
```

TRACE SQL and SHOW TRACE

You can run these SQL statements to view the execution results of an SQL statement. Note that you must use TRACE SQL and SHOW TRACE together. The difference between TRACE SQL and EXPLAIN SQL is that TRACE SQL is truly executed.

For example, you can run these statements to view the execution results of the select 1 statement.

```
mysql> trace select 1;
+---+
| 1 |
+---+
| 1 |
+---+
1 row in set (0.03 sec)
mysql> show trace;
+------+----------+----------------+----------------------------------+---------------+-------------
-------------+------+-----------+---------+
| ID   | TYPE     | GROUP_NAME     | DBKEY_NAME                       | TIME_COST(MS) | CONNECTION_T
IME_COST(MS) | ROWS | STATEMENT | PARAMS |
+------+----------+----------------+----------------------------------+---------------+-------------
-------------+------+-----------+---------+
|    0 | Optimize | DRDS           | DRDS                             | 3             | 0.00
|    0 | select 1  | NULL   |
|    1 | Query    | TDDL5_00_GROUP | db218249098_sqa_zmf_tddl5_00_3309 | 7             | 0.15
|    1 | select 1  | NULL   |
+------+----------+----------------+----------------------------------+---------------+-------------
-------------+------+-----------+---------+
2 rows in set (0.01 sec)
```

CHECK TABLE tablename

You can run this SQL statement to check a data table. This SQL statement is mainly used when a table failed to be created by using a data definition language (DDL) statement.

- If the data table is a table shard, this SQL statement allows you to check whether any underlying physical table shard is missing and whether the columns and indexes of the underlying physical table are consistent.
- If the data table is a single-database non-partition table, this SQL statement allows you to check whether this table exists.

```
mysql>  check table tddl_mgr_log;
+------------------------+-------+----------+----------+
| TABLE                  | OP    | MSG_TYPE | MSG_TEXT |
+------------------------+-------+----------+----------+
| TDDL5_APP.tddl_mgr_log | check | status   | OK       |
+------------------------+-------+----------+----------+
1 row in set (0.56 sec)
mysql> check table tddl_mg;
+-------------------+-------+----------+----------------------------------------+
| TABLE             | OP    | MSG_TYPE | MSG_TEXT                               |
+-------------------+-------+----------+----------------------------------------+
| TDDL5_APP.tddl_mg | check | Error    | Table 'tddl5_00.tddl_mg' doesn't exist |
+-------------------+-------+----------+----------------------------------------+
1 row in set (0.02 sec)
```

SHOW TABLE STATUS [LIKE 'pattern' | WHERE expr]

You can run this SQL statement to obtain the information about a table. This command aggregates the data of all underlying physical table shards.

The following describes the meanings of important columns:

- NAME: indicates the name of the table.
- ENGINE: indicates the storage engine of the table.
- VERSION: indicates the version of the storage engine of the table.
- ROW_FORMAT: indicates the format of the rows in the table. Valid values include Dynamic, Fixed, and Compressed. The value Dynamic indicates that the row length is variable, for example, is a VARCHAR or BLOB field. The value Fixed indicates that the row length is constant, for example, is a CHAR or INTEGER field.
- ROWS: indicates the number of rows in the table.
- AVG_ROW_LENGTH: indicates the average number of bytes in each row.
- DATA_LENGTH: indicates the data volume of the entire table, in bytes.
- MAX_DATA_LENGTH: indicates the maximum volume of data that can be stored in the table.
- INDEX_LENGTH: indicates the size of the disk space occupied by indexes.
- CREATE_TIME: indicates the time when the table was created.
- UPDATE_TIME: indicates the time when the table was last updated.
- COLLATION: indicates the default character set and character sorting rule of the table.
- CREATE_OPTIONS: indicates all the other options specified when the table was created.

```
mysql> show table status like 'multi_db_multi_tbl';
+--------------------+--------+---------+------------+------+----------------+-------------+---------
--------+--------------+-----------+----------------+----------------------+-------------+----------
--+-----------------+----------+----------------+---------+
| NAME               | ENGINE | VERSION | ROW_FORMAT | ROWS | AVG_ROW_LENGTH | DATA_LENGTH | MAX_DATA
_LENGTH | INDEX_LENGTH | DATA_FREE | AUTO_INCREMENT | CREATE_TIME          | UPDATE_TIME | CHECK_TIM
E | COLLATION       | CHECKSUM | CREATE_OPTIONS | COMMENT |
+--------------------+--------+---------+------------+------+----------------+-------------+---------
--------+--------------+-----------+----------------+----------------------+-------------+----------
--+-----------------+----------+----------------+---------+
| multi_db_multi_tbl | InnoDB |      10 | Compact    |    2 |          16384 |       16384 |
0 |        16384 |         0 |         100000 | 2017-03-27 17:43:57.0 | NULL        | NULL       | ut
f8_general_ci | NULL     |                |         |
+--------------------+--------+---------+------------+------+----------------+-------------+---------
--------+--------------+-----------+----------------+----------------------+-------------+----------
--+-----------------+----------+----------------+---------+
1 row in set (0.03 sec)
```

The combination of the SHOW TABLE STATUS statement and the SCAN hint allows you to view the data volume of each physical table shard.

```
mysql> /! TDDL:SCAN='multi_db_multi_tbl'*/show table status like 'multi_db_multi_tbl';
+----------------------+--------+---------+------------+------+----------------+-------------+-------
----------+--------------+-----------+----------------+---------------------+-------------+----------
--+-----------------+----------+----------------+---------+--------------+
| Name                 | Engine | Version | Row_format | Rows | Avg_row_length | Data_length | Max_da
ta_length | Index_length | Data_free | Auto_increment | Create_time         | Update_time | Check_tim
e | Collation       | Checksum | Create_options | Comment | Block_format |
+----------------------+--------+---------+------------+------+----------------+-------------+-------
----------+--------------+-----------+----------------+---------------------+-------------+----------
--+-----------------+----------+----------------+---------+--------------+
| multi_db_multi_tbl_1 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_0 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
```

```
| multi_db_multi_tbl_1 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_0 | InnoDB |      10 | Compact    |    1 |          16384 |       16384 |
0 |        16384 |         0 |              2 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_1 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_0 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_1 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_0 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_1 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_0 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_1 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_0 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_1 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_0 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_1 | InnoDB |      10 | Compact    |    0 |              0 |       16384 |
0 |        16384 |         0 |              1 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
| multi_db_multi_tbl_0 | InnoDB |      10 | Compact    |    1 |          16384 |       16384 |
0 |        16384 |         0 |              3 | 2017-03-27 17:43:57 | NULL        | NULL       | utf8
_general_ci |     NULL |                |         | Original     |
+----------------------+---------+--------+-----------+-------+----------------+--------------+------
----------+--------------+------------+----------------+---------------------+-------------+----------
--+-----------------+----------+----------------+----------+--------------+
16 rows in set (0.04 sec)
```

13.1.7.5. Statistics query statements

SHOW [FULL] STATS

You can run this SQL statement to view the overall statistics of a Distributed Relational Database Service (DRDS) instance. The statistics are instantaneous values. Note that the results of `DRDS SHOW FULL STATS` vary with the version of the DRDS instance.

The following describes the meanings of important columns:

- QPS: the number of queries per second (QPS) sent from an application to the DRDS instance. These queries are

usually called logical QPS.

- RDS_QPS: the number of QPS sent from the DRDS instance to an ApsaraDB RDS for MySQL instance. These queries are usually called physical QPS.
- ERROR_PER_SECOND: the total number of errors that occur on the DRDS instance per second. These errors include various errors such as SQL syntax errors, primary key conflicts, system errors, and connectivity errors.
- VIOLATION_PER_SECOND: the number of conflicts that occur on primary keys or unique keys per second.
- MERGE_QUERY_PER_SECCOND: the number of queries processed on multiple tables through database sharding and table sharding per second.
- ACTIVE_CONNECTIONS: the number of active connections to the DRDS instance.
- CONNECTION_CREATE_PER_SECCOND: the number of connections that are created for the DRDS instance per second.
- RT(MS): the time to respond to an SQL query sent from an application to the DRDS instance. This response time (RT) is usually called logical RT.
- RDS_RT(MS): the time to respond to an SQL query sent from the DRDS instance to an ApsaraDB RDS for MySQL instance. This RT is usually called physical RT.
- NET_IN(KB/S): the amount of inbound traffic of the DRDS instance per second.
- NET_OUT(KB/S): the amount of outbound traffic of the DRDS instance per second.
- THREAD_RUNNING: the number of threads that are running in the DRDS instance.
- HINT_USED_PER_SECOND: the number of SQL queries that contain hints and are processed by the DRDS instance per second.
- HINT_USED_COUNT: the total number of SQL queries that contain hints and have been processed by the DRDS instance since startup.
- AGGREGATE_QUERY_PER_SECCOND: the number of aggregate SQL queries processed by the DRDS instance per second.
- AGGREGATE_QUERY_COUNT: the total number of aggregate SQL queries that have been processed by the DRDS instance.
- TEMP_TABLE_CREATE_PER_SECCOND: the number of temporary tables created in the DRDS instance per second.
- TEMP_TABLE_CREATE_COUNT: the total number of temporary tables that have been created in the DRDS instance since startup.
- MULTI_DB_JOIN_PER_SECCOND: the number of multi-database JOIN queries processed by the DRDS instance per second.
- MULTI_DB_JOIN_COUNT: the number of multi-database JOIN queries that have been processed by the DRDS instance since startup.

```
mysql> show stats;
+------+---------+----------+------------------+-----------------+------------------------+--------
------------+--------+------------+--------------+---------------+----------------+
| QPS  | RDS_QPS | SLOW_QPS | PHYSICAL_SLOW_QPS | ERROR_PER_SECOND | MERGE_QUERY_PER_SECOND | ACTIVE_
CONNECTIONS | RT(MS) | RDS_RT(MS) | NET_IN(KB/S) | NET_OUT(KB/S) | THREAD_RUNNING |
+------+---------+----------+------------------+-----------------+------------------------+--------
------------+--------+------------+--------------+---------------+----------------+
| 1.77 |    1.68 |     0.03 |              0.03 |             0.02 |                   0.00 |
7 | 157.13 |      51.14 |       134.49 |          1.48 |              1 |
+------+---------+----------+------------------+-----------------+------------------------+--------
------------+--------+------------+--------------+---------------+----------------+
1 row in set (0.01 sec)
mysql> show full stats;
+------+---------+----------+------------------+-----------------+------------------------+----------
-------------+--------------------+-------------------------------+--------+------------+------------
--+---------------+----------------+----------------------+-----------------+------------------------
----+-----------------------+-------------------------------+-------------------------+---------------
-----------+---------------------+-------+---------+-------------+-------------+
| QPS  | RDS_QPS | SLOW_QPS | PHYSICAL_SLOW_QPS | ERROR_PER_SECOND | VIOLATION_PER_SECOND | MERGE_QUE
RY_PER_SECOND | ACTIVE_CONNECTIONS | CONNECTION_CREATE_PER_SECOND | RT(MS) | RDS_RT(MS) | NET_IN(KB/S
) | NET_OUT(KB/S) | THREAD_RUNNING | HINT_USED_PER_SECOND | HINT_USED_COUNT | AGGREGATE_QUERY_PER_SEC
OND | AGGREGATE_QUERY_COUNT | TEMP_TABLE_CREATE_PER_SECOND | TEMP_TABLE_CREATE_COUNT | MULTI_DB_JOIN_
PER_SECOND | MULTI_DB_JOIN_COUNT | CPU   | FREEMEM | FULLGCCOUNT | FULLGCTIME |
+------+---------+----------+------------------+-----------------+------------------------+----------
-------------+--------------------+-------------------------------+--------+------------+------------
--+---------------+----------------+----------------------+-----------------+------------------------
----+-----------------------+-------------------------------+-------------------------+---------------
-----------+---------------------+-------+---------+-------------+-------------+
| 1.63 |    1.68 |     0.03 |              0.03 |             0.02 |                   0.00 |
0.00 |                   6 |                          0.01 | 157.13 |      51.14 |       134.33 |
1.21 |              1 |                 0.00 |              54 |                         0.00 |
663 |                          0.00 |                     512 |                      0.00 |
516 | 0.09% |   6.96% |       76446 |   21326906 |
+------+---------+----------+------------------+-----------------+------------------------+----------
-------------+--------------------+-------------------------------+--------+------------+------------
--+---------------+----------------+----------------------+-----------------+------------------------
----+-----------------------+-------------------------------+-------------------------+---------------
-----------+---------------------+-------+---------+-------------+-------------+
1 row in set (0.01 sec)
```

SHOW DB STATUS

You can run this SQL statement to view the capacity and performance information of a physical database, which is also called a database shard. All the returned values indicate the real-time information. The capacity information is obtained from the ApsaraDB RDS for MySQL system table, and therefore may be different from the actual capacity information.

The following describes the meanings of important columns:

- NAME: the internal tag that represents a logical database corresponding to the database shard. The value is different from the name of the logical database.
- CONNECTION_STRING: the information about a connection from the DRDS instance to the database shard.
- PHYSICAL_DB: the name of the database shard. The `TOTAL` row indicates the total amount of capacity of all the database shards corresponding to the logical database.
- SIZE_IN_MB: the size of the space occupied by the data in the database shard. Unit: MB
- RATIO: the ratio of the data volume of the database shard to the total data volume of the current logical database.

- THREAD_RUNNING: the number of threads that are running in the ApsaraDB RDS for MySQL instance to which the physical database belongs. The meaning of this parameter is the same as that of the THREAD_RUNNING parameter returned by the MySQL `SHOW GLOBAL STATUS` command. For more information, see MySQL Documentation.

```
mysql> show db status;
+------+---------------------------+-------------------+------------------+------------+--------+--
--------------+
| ID   | NAME                      | CONNECTION_STRING  | PHYSICAL_DB       | SIZE_IN_MB | RATIO  | T
HREAD_RUNNING |
+------+---------------------------+-------------------+------------------+------------+--------+--
--------------+
|    1 | drds_db_1516187088365daui | 100.100.64.1:59077 | TOTAL             |  13.109375 | 100%   | 3
|
|    2 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0000 |   1.578125 | 12.04% |
|
|    3 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0001 |     1.4375 | 10.97% |
|
|    4 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0002 |     1.4375 | 10.97% |
|
|    5 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0003 |     1.4375 | 10.97% |
|
|    6 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0004 |   1.734375 | 13.23% |
|
|    7 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0005 |   1.734375 | 13.23% |
|
|    8 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0006 |   2.015625 | 15.38% |
|
|    9 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0007 |   1.734375 | 13.23% |
|
+------+---------------------------+-------------------+------------------+------------+--------+--
--------------+
```

SHOW FULL DB STATUS [LIKE {tablename}]

You can run this SQL statement to view the capacity and performance information of a table shard in a physical database, which is also called a database shard. All the returned values indicate the real-time information. The capacity information is obtained from the ApsaraDB RDS for MySQL system table, and therefore may be different from the actual capacity information.

The following describes the meanings of important columns:

- NAME: the internal tag that represents a logical database corresponding to the database shard. The value is different from the name of the logical database.
- CONNECTION_STRING: the information about a connection from the DRDS instance to the database shard.
- PHYSICAL_DB: the name of the database shard. If the LIKE keyword is used for filtering in the statement, the TOTAL row indicates the total amount of capacity of the database shard. If the LIKE keyword is not used for filtering in the statement, the TOTAL row indicates the total amount of capacity of all database shards.
- PHYSICAL_TABLE: the name of the table shard in the database shard. If the LIKE keyword is used for filtering in the statement, the TOTAL row indicates the total amount of capacity of the table shard. If the LIKE keyword is not used for filtering in the statement, the TOTAL row indicates the total amount of capacity of all table shards in the database shard.
- SIZE_IN_MB: the size of the space occupied by the data in the database shard. Unit: MB
- RATIO: the ratio of the data volume of the table shard to the total data volume of all the table shards obtained through filtering.
- THREAD_RUNNING: the number of threads that are running in the ApsaraDB RDS for MySQL instance to which the physical database belongs. The meaning of this parameter is the same as that of the THREAD_RUNNING

parameter returned by the MySQL `SHOW GLOBAL STATUS` command. For more information, see MySQL Documentation.

```
mysql> show full db status like hash_tb;
+------+--------------------------+-------------------+------------------+----------------+-------
------+--------+----------------+
| ID   | NAME                     | CONNECTION_STRING  | PHYSICAL_DB       | PHYSICAL_TABLE | SIZE_I
N_MB | RATIO  | THREAD_RUNNING |
+------+--------------------------+-------------------+------------------+----------------+-------
------+--------+----------------+
|    1 | drds_db_1516187088365daui | 100.100.64.1:59077 | TOTAL             |                |     19
.875 | 100%   | 3              |
|    2 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0000 | TOTAL          |    3.0
3125 | 15.25% |                |
|    3 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0000 | hash_tb_00     |   1.51
5625 | 7.63%  |                |
|    4 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0000 | hash_tb_01     |   1.51
5625 | 7.63%  |                |
|    5 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0001 | TOTAL          |
2.0 | 10.06% |                |
|    6 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0001 | hash_tb_02     |   1.51
5625 | 7.63%  |                |
|    7 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0001 | hash_tb_03     |   0.48
4375 | 2.44%  |                |
|    8 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0002 | TOTAL          |    3.0
3125 | 15.25% |                |
|    9 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0002 | hash_tb_04     |   1.51
5625 | 7.63%  |                |
|   10 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0002 | hash_tb_05     |   1.51
5625 | 7.63%  |                |
|   11 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0003 | TOTAL          |   1.95
3125 | 9.83%  |                |
|   12 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0003 | hash_tb_06     |   1.51
5625 | 7.63%  |                |
|   13 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0003 | hash_tb_07     |     0.
4375 | 2.2%   |                |
|   14 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0004 | TOTAL          |    3.0
3125 | 15.25% |                |
|   15 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0004 | hash_tb_08     |   1.51
5625 | 7.63%  |                |
|   16 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0004 | hash_tb_09     |   1.51
5625 | 7.63%  |                |
|   17 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0005 | TOTAL          |   1.92
1875 | 9.67%  |                |
|   18 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0005 | hash_tb_11     |   1.51
5625 | 7.63%  |                |
|   19 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0005 | hash_tb_10     |    0.4
0625 | 2.04%  |                |
|   20 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0006 | TOTAL          |    3.0
3125 | 15.25% |                |
|   21 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0006 | hash_tb_12     |   1.51
5625 | 7.63%  |                |
|   22 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0006 | hash_tb_13     |   1.51
5625 | 7.63%  |                |
|   23 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0007 | TOTAL          |      1
.875 | 9.43%  |                |
|   24 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0007 | hash_tb_14     |   1.51
5625 | 7.63%  |                |
|   25 | drds_db_1516187088365daui | 100.100.64.1:59077 | drds_db_xzip_0007 | hash_tb_15     |   0.35
```

```
9375 | 1.81%  |                  |
+------+--------------------------+--------------------+------------------+----------------+-------
-----+--------+----------------+
```

13.1.7.6. SHOW PROCESSLIST and KILL commands

> **Note**
>
> - If the version of is 5.1.28-1408022 or later, support the SHOW PROCESSLIST and KILL commands for both logical and physical connections. For more information, see this topic.
> - If the version of is earlier than 5.1.28-1408022, support the SHOW PROCESSLIST and KILL commands only for physical connections. For more information, see SHOW PROCESSLIST and KILL commands in earlier versions.

SHOW PROCESSLIST command

In a instance, you can run the `SHOW PROCESSLIST` command to view information such as connections to the instance and SQL statements that are being executed in the PolarDB-X instance.

Syntax

```
SHOW [FULL] PROCESSLIST
```

Examples

```
mysql> SHOW PROCESSLIST\G
     ID: 1971050
   USER: admin
   HOST: 111.111.111.111:4303
     DB: drds_test
COMMAND: Query
   TIME: 0
  STATE:
   INFO: show processlist
1 row in set (0.01 sec)
```

The following describes the meanings of the fields in the result set:

- ID: the ID of the connection. The value is a long-type number.
- USER: the name of the user who sets up the connection.
- HOST: the IP address and port number of the host that sets up the connection.
- DB: the name of the database accessed by the connection.
- COMMAND: the usage state of the connection. Currently, this field can be set to the following values:
  - Query: the current connection is executing an SQL statement.
  - Sleep: the current connection is idle.
- TIME: the duration when the connection is in the current state:
  - When the value of COMMAND is Query, this field indicates how long the SQL statement has been being executed on the connection.
  - When the value of COMMAND is Sleep, this field indicates how long the connection has been in the idle state.
- STATE: currently, no meaning has been assigned for this field. The value is constantly empty.
- INFO:

- When the value of COMMAND is Query, this field indicates the content of the SQL statement that is being executed on the connection. If the FULL parameter is not specified, a maximum of the first 30 characters of the SQL statement are returned. If the FULL parameter is specified, a maximum of the first 1000 characters of the SQL statement are returned.
- When the value of COMMAND is other values, this field is meaningless and left empty.

SHOW PHYSICAL_PROCESSLIST command

In a instance, you can run the SHOW PHYSICAL_PROCESSLIST command to view information about all the SQL statements that are being executed on underlying ApsaraDB RDS for MySQL instances.

Syntax

```
SHOW [FULL] PHYSICAL_PROCESSLIST
```

When an SQL statement is excessively long, the responses of the `SHOW PHYSICAL_PROCESSLIST` command may be truncated. In this case, you can run the `SHOW FULL PHYSICAL_PROCESSLIST` command to obtain the complete SQL statement.

The meaning of each column in the responses is equivalent to that in the responses of the `SHOW PROCESSLIST` command. For more information, see SHOW PROCESSLIST Syntax.

> **Note** Different from ApsaraDB RDS for MySQL, the instance returns a string instead of a number in the ID column of a physical connection.

```
mysql> SHOW PHYSICAL_PROCESSLIST\G
*************************** 1. row ***************************
           ID: 0-0-521414
         USER: tddl5
           DB: tddl5_00
      COMMAND: Query
         TIME: 0
        STATE: init
         INFO: show processlist
*************************** 2. row ***************************
           ID: 0-0-521570
         USER: tddl5
           DB: tddl5_00
      COMMAND: Query
         TIME: 0
        STATE: User sleep
         INFO: /*DRDS /88.88.88.88/b67a0e4d8800000/ */ select sleep(1000)
2 rows in set (0.01 sec)
```

KILL command

The KILL command is used to terminate an SQL statement that is being executed.

The instance connects to an ApsaraDB RDS for MySQL instance by using the username created by the instance on the ApsaraDB RDS for MySQL instance. Therefore, if you directly connect to the ApsaraDB RDS for MySQL instance, you do not have the permission to run the KILL command on a request initiated by the instance.

To terminate an SQL statement that is being executed on the instance, you must use tools such as the MySQL command line and to connect to the instance, and then run the KILL command on the instance.

Syntax

```
KILL PROCESS_ID | 'PHYSICAL_PROCESS_ID' | 'ALL'
```

The KILL command can be used in the following ways:

- Run `KILL PROCESS_ID` to terminate a specified logical SQL statement.

  The PROCESS_ID parameter is obtained from the ID column in the responses of the `SHOW [FULL] PROCESSLIST` command.

  Running the `KILL PROCESS_ID` command in the instance will terminate both logical and physical SQL statements that are being executed on this connection, and disconnect this connection.

  The instance does not support the `KILL QUERY` command.

- Run `KILL 'PHYSICAL_PROCESS_ID'` to terminate a specified physical SQL statement.

  The PHYSICAL_PROCESS_ID parameter is obtained from the ID column in the responses of the `SHOW PHYSICAL_PROCESS_ID` command.

  > **Note** The PHYSICAL_PROCESS_ID column is a string instead of a number. Therefore, the PHYSICAL_PROCESS_ID parameter must be enclosed in single quotation marks (') in the KILL command.

  Examples:

  ```
  mysql> KILL '0-0-521570';
  Query OK, 0 rows affected (0.01 sec)
  ```

- Run `KILL 'ALL'` to terminate all the physical SQL statements that are executed by the instance in the current logical database.

  When the underlying ApsaraDB RDS for MySQL instance is overloaded due to some SQL statements, you can use the `KILL 'ALL'` command to terminate all the physical SQL statements that are being executed in the current logical database.

  All physical SQL statements indicated by PROCESS that meet the following conditions can be terminated by running `KILL 'ALL'`:

  - The value of the User parameter for the physical SQL statement indicated by PROCESS is a username created by the instance in the ApsaraDB RDS for MySQL instance.
  - The physical SQL statement indicated by PROCESS is executing a query. In other words, the value of COMMAND is Query.

13.1.7.7. SHOW PROCESSLIST and KILL commands in earlier versions

> **Note**
>
> - If the version of is 5.1.28-1408022 or later, support the SHOW PROCESSLIST and KILL commands for both logical and physical connections. For more information, see SHOW PROCESSLIST and KILL commands.
> - If the version of is earlier than 5.1.28-1408022, only supports the SHOW PROCESSLIST and KILL commands for physical connections. For more information, see this topic.

SHOW PROCESSLIST command

In a instance, you can run the `SHOW PROCESSLIST` command to view information about all the SQL statements that are being executed on the ApsaraDB RDS for MySQL instances.

Syntax

```
SHOW [FULL] PROCESSLIST
```

When an SQL statement is excessively long, the responses of the `SHOW PROCESSLIST` command may be truncated. In this case, you can run the `SHOW FULL PROCESSLIST` command to obtain the complete SQL statement.

The meaning of each column in the responses is equivalent to that in the responses of the `SHOW PROCESSLIST` statement. For more information, see SHOW PROCESSLIST Syntax.

```
mysql> SHOW PROCESSLIST\G
*************************** 1. row ***************************
           ID: 0-0-521414
         USER: tddl5
           DB: tddl5_00
      COMMAND: Query
         TIME: 0
        STATE: init
         INFO: show processlist
    ROWS_SENT: NULL
ROWS_EXAMINED: NULL
    ROWS_READ: NULL
*************************** 2. row ***************************
           ID: 0-0-521570
         USER: tddl5
           DB: tddl5_00
      COMMAND: Query
         TIME: 0
        STATE: User sleep
         INFO: /*DRDS /88.88.88.88/b67a0e4d8800000/ */ select sleep(1000)
    ROWS_SENT: NULL
ROWS_EXAMINED: NULL
    ROWS_READ: NULL
2 rows in set (0.01 sec)
```

KILL

You can execute the KILL statement to terminate an SQL statement that is being executed.

The instance connects to an ApsaraDB RDS for MySQL instance by using the username created by the instance on the ApsaraDB RDS for MySQL instance. Therefore, if you directly connect to the ApsaraDB RDS for MySQL instance, you are not authorized to execute the KILL statement to terminate a request initiated by the instance.

To terminate an SQL statement that is being executed on the instance, you must use tools to connect to the instance. You can use tools such as the MySQL command line and . Then, execute the KILL statement on the instance.

Syntax

```
KILL 'PROCESS_ID' | 'ALL'
```

The KILL command can be used in the following ways:

- Run `KILL 'PROCESS_ID'` to terminate a specified SQL statement.

  The PROCESS_ID parameter is obtained from the ID column in the responses of the `SHOW PROCESSLIST` command.

> **Note** Different from ApsaraDB RDS for MySQL, the instance returns a string instead of a number in the ID column. Therefore, the PROCESS_ID parameter must be enclosed in single quotation marks (') in the KILL command.

Examples

```
mysql> KILL '0-0-521570';
Query OK, 0 rows affected (0.01 sec)
```

- Run `KILL 'ALL'` to terminate all the SQL statements executed by the instance in the current logical database.

  When the underlying ApsaraDB RDS for MySQL instance is overloaded due to several SQL statements, you can use the `KILL 'ALL'` command to terminate all the SQL statements that are being executed in the current logical database.

  All SQL statements indicated by PROCESS that meet the following conditions can be terminated by running `KILL 'ALL'`:

  - The value of the User parameter for the physical SQL statement indicated by PROCESS is a username created by the instance in the ApsaraDB RDS for MySQL instance.
  - The physical SQL statement indicated by PROCESS is executing a query, which means that the value of COMMAND is Query.

  instances in earlier versions do not support the `KILL 'ALL'` command. An error will be reported if this command is being executed in these instances. To resolve this problem, you can upgrade the version of the instance.

13.1.8. Custom hints

> **Note** This topic is applicable to 5.3 and later. For earlier versions, see PolarDB-X 5.2 hints.

13.1.8.1. Introduction to hints

As a supplement to the SQL syntax, hints play a critical role in relational databases. They allow you to influence execution plans of SQL statements by using relevant syntax, to specially optimize the SQL statements. also provides special hint syntax.

For example, if you know the target data is stored in table shards in certain database shards and you need to route the SQL statement directly to the database shards for execution, you can use custom hints provided by .

```
SELECT /*+TDDL:node('node_name')*/ * FROM table_name;
```

In the preceding SQL statement, the part between `/*` and `*/`, namely, `+TDDL:node('node_name')`, is a hint. The hint specifies the ApsaraDB RDS for MySQL database shard where the SQL statement is to be executed.

> **Note**
> - hints can be in the formats of `/*+TDDL:hint_command*/` and `/! +TDDL:hint_command*/`.
> - In the MySQL command-line client, if you need to run an SQL statement that contains a hint in the format of `/*+TDDL:hint_command*/`, add the `-c` parameter to the logon command, because hints are based on the MySQL Comment Syntax. Otherwise, the client deletes the hint and then sends the SQL statement to the server for execution, which causes the hint to fail to take effect. For more information, see MySQL Client Options.

PolarDB-X hint syntax

Basic syntax

```
/*+TDDL: hint_command [hint_command ...] */
/! +TDDL: hint_command [hint_command ...] */
```

hints are based on the MySQL Comment Syntax. The hint statements are located between `/*` and `*/` or between `/!` and `*/`, and must begin with `+TDDL:`. The `hint_command` parameter indicates a hint command related to the specific operation. Multiple `hint_command` parameters are separated by spaces.

Examples

```
# Query the names of physical tables in each database shard.
/*+TDDL:scan()*/SHOW TABLES;
# Route the query to database shard 0000 of a read-only ApsaraDB RDS for MySQL instance.
/*+TDDL:node(0) slave()*/SELECT * FROM t1;
```

In the example, `/*+TDDL:scan()*/` and `/*+TDDL:node(0) slave()*/` are hints that begin with `+TDDL:`. The `scan()`, `node(0)`, and `slave()` functions are hint commands. Hint commands are separated by spaces.

- Use one hint in an SQL statement:

  allows you to use hints in data manipulation language (DML), data definition language (DDL), and data access language (DAL) statements. The following describes the syntax in detail.

  - For all statements that support hints, you can specify a hint at the beginning of the statements, for example:

    ```
    /*+TDDL: ... */ SELECT ...
    /*+TDDL: ... */ INSERT ...
    /*+TDDL: ... */ REPLACE ...
    /*+TDDL: ... */ UPDATE ...
    /*+TDDL: ... */ DELETE ...
    /*+TDDL: ... */ CREATE TABLE ...
    /*+TDDL: ... */ ALTER TABLE ...
    /*+TDDL: ... */ DROP TABLE ...
    /*+TDDL: ... */ SHOW ...
    ...
    ```

  - For DML statements, you can specify a hint behind the first keyword of the statements, for example:

    ```
    SELECT /*+TDDL: ... */  ...
    INSERT /*+TDDL: ... */  ...
    REPLACE /*+TDDL: ... */  ...
    UPDATE /*+TDDL: ... */  ...
    DELETE /*+TDDL: ... */  ...
    ...
    ```

    Note Different hints may be applicable to different syntaxes. For more information about the applicable syntaxes, see the documentation of hint commands.

- Use multiple hint commands in an SQL statement:

  allows you to use multiple hint commands in SQL statements that contain hints.

  ```
  SELECT /*+TDDL:node(0) slave()*/ ... ;
  ```

  has the following limitations on the use of multiple hint commands:

```
# A single SQL statement cannot contain multiple hint statements.
SELECT /*+TDDL:node(0)*/ /*+TDDL:slave()*/ ... ;
# An SQL statement that contains a hint cannot contain duplicate hint commands.
SELECT /*+TDDL:node(0) node(1)*/ ... ;
```

PolarDB-X hint classification

hints are classified into the following major categories according to operation types:

- Read/write splitting
- Specify a timeout period for an SQL statement
- Specify a database shard to run an SQL statement
- Scan all or some of database shards and table shards

13.1.8.2. Read/write splitting

provides transparent read/write splitting at the application layer. Data synchronization between primary and read-only ApsaraDB RDS for MySQL instances has a delay of several milliseconds. If you need to read changed data immediately after the primary ApsaraDB RDS for MySQL instance is changed, you must ensure that the SQL statement for reading data is routed to the primary ApsaraDB RDS for MySQL instance. To meet this demand, provides custom hints for read/write splitting, to route SQL statements to a specified primary or read-only ApsaraDB RDS for MySQL instance.

> **Note** This topic is applicable to 5.3 and later. For earlier versions, see Read/write splitting.

Syntax

```
/*+TDDL:
    master()
    | slave()
*/
```

With this custom hint, you can specify whether to run an SQL statement on a primary or read-only ApsaraDB RDS for MySQL instance. With the custom hint `/*+TDDL:slave()*/`, if a primary ApsaraDB RDS for MySQL instance is configured with multiple read-only ApsaraDB RDS for MySQL instances, the instance randomly selects a read-only ApsaraDB RDS for MySQL instance based on its weight, to run the SQL statement.

> **Note**
> - hints can be in the formats of `/*+TDDL:hint_command*/` and `/! +TDDL:hint_command*/`.
> - In the MySQL command-line client, you may need to execute an SQL statement that contains a hint in the format of `/*+TDDL:hint_command*/`. In this case, add the `-c` parameter to the logon command. Otherwise, the client deletes the hint before it sends the SQL statement to the server for execution because the hint is in the format of a MySQL comment. In this case, the hint fails to take effect. For more information, see MySQL Client Options.

Examples

- Specify a primary ApsaraDB RDS for MySQL instance to run an SQL statement:

```
SELECT /*+TDDL:master()*/ * FROM table_name;
```

After the custom hint `/*+TDDL:master()*/` is added behind the first keyword in the SQL statement, this SQL statement is routed to the primary ApsaraDB RDS for MySQL instance for execution.

- Specify a read-only ApsaraDB RDS for MySQL instance to run an SQL statement:

```
SELECT /*+TDDL:slave()*/ * FROM table_name;
```

After the custom hint `/*+TDDL:slave()*/` is added behind the first keyword in the SQL statement, this SQL statement is randomly routed to a read-only ApsaraDB RDS for MySQL instance based on the allocated weight.

Note

- The custom hints for read-write splitting are only applicable to read SQL statements for non-transactional data. SQL statements for transactional data and write SQL statements are still routed to the primary ApsaraDB RDS for MySQL instance for execution.
- The hint `/*+TDDL:slave()*/` allows you to route the SQL statement randomly to a read-only ApsaraDB RDS for MySQL instance based on the configured weight for execution. If no read-only ApsaraDB RDS for MySQL instance is available, no error is reported. Instead, the primary ApsaraDB RDS for MySQL instance is selected to run the SQL statement.

13.1.8.3. Specify a timeout period for an SQL statement

In , the SQL statements for instances and ApsaraDB RDS for MySQL instances are timed out after 900 seconds (which can be adjusted) by default. However, for some slow SQL statements, the execution duration may exceed 900 seconds. For these slow SQL statements, provides a custom hint to adjust their timeout periods. You can use this custom hint to adjust the SQL execution duration as needed.

> **Note** This topic is applicable to 5.3 and later. For earlier versions, see Specify a timeout period for an SQL statement.

Syntax

The syntax of the hint for specifying a timeout period for an SQL statement is as follows:

```
/*+TDDL:SOCKET_TIMEOUT(time)*/
```

The `SOCKET_TIMEOUT` parameter is measured in milliseconds. With this custom hint, you can adjust the timeout period for the SQL statement based on business requirements.

> **Note**
> - hints can be in the formats of `/*+TDDL:hint_command*/` and `/! +TDDL:hint_command*/`.
> - In the MySQL command-line client, you may need to execute an SQL statement that contains a hint in the format of `/*+TDDL:hint_command*/`. In this case, add the `-c` parameter to the logon command. Otherwise, the client deletes the hint before it sends the SQL statement to the server for execution because the hint is in the format of a MySQL comment. In this case, the hint fails to take effect. For more information, see MySQL Client Options.

Examples

Set the timeout period of an SQL statement to 40 seconds:

```
/*+TDDL:SOCKET_TIMEOUT(40000)*/SELECT * FROM t_item;
```

> **Note** A longer timeout period causes database resources to be occupied for a longer period of time. If excessive SQL statements are executed over a long time within the same period, a large number of database resources may be consumed. This will make users unable to use PolarDB-X properly. In this case, we need to use this custom hint to optimize the SQL statements that take a long time to execute.

13.1.8.4. Specify a database shard to run an SQL statement

When running SQL commands in a instance, you may find that some SQL statements are not supported by the instance. In this case, you can use the `NODE HINT` provided by , to route the SQL statements to one or more database shards for execution. In addition, if you need to query the data in a specified database shard or the data in a specified table shard in a known database shard, you can use the `NODE HINT` to directly route the SQL statement to the database shard for execution.

> **Note** This topic is applicable to 5.3 and later. For earlier versions, see Specify a database shard to run an SQL statement.

Syntax

The `NODE HINT` allows you to specify a database shard by using a shard name, to run the SQL statement in the database shard. A shard name uniquely identifies a database shard in a instance. You can run the SHOW NODE statement to obtain the shard name.

Specify a database shard by using a shard name, to run an SQL statement

This custom hint allows you to specify one or more database shards to run an SQL statement.

> **Note** If the hint for specifying a database shard is used in an INSERT statement that contains a sequence for the target table, the sequence will not take effect. For more information, see Limits and precautions for sequences.

- Specify one database shard to run an SQL statement:

  ```
  /*+TDDL:node('node_name')*/
  ```

  Specifically, `node_name` indicates the shard name. This hint enables you to route the SQL statement to the database shard specified by `node_name`.

- Specify multiple database shards to run an SQL statement:

  ```
  /*+TDDL:node('node_name'[,'node_name1','node_name2'])*/
  ```

  You can specify multiple shard names in the parameters and route the SQL statement to multiple database shards for execution. Separate multiple shard names with commas (,).

Note

- When this custom hint is used, the instance directly routes the SQL statement to the specified database shards for execution. Therefore, the specified shard names in the SQL statement must correspond to existing database shards.
- The `NODE HINT` can be used in data manipulation language (DML), data definition language (DDL), and data access language (DAL) statements.
- hints can be in the formats of `/*+TDDL:hint_command*/` and `/! +TDDL:hint_command*/`.
- In the MySQL command-line client, you may need to execute an SQL statement that contains a hint in the format of `/*+TDDL:hint_command*/`. In this case, add the `-c` parameter to the logon command. Otherwise, the client deletes the hint before it sends the SQL statement to the server for execution because the hint is in the format of a MySQL comment. In this case, the hint fails to take effect. For more information, see MySQL Client Options.

Examples

The following shows the responses of the SHOW NODE statement for a logical database named `drds_test` in a instance.

```
mysql> SHOW NODE\G
*************************** 1. row ******************
                ID: 0
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS
  MASTER_READ_COUNT: 212
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 2. row ******************
                ID: 1
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0001_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 3. row ******************
                ID: 2
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0002_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 4. row ******************
                ID: 3
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0003_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 5. row ******************
                ID: 4
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0004_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 6. row ******************
                ID: 5
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0005_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 7. row ******************
                ID: 6
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0006_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 8. row ******************
                ID: 7
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0007_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
8 rows in set (0.02 sec)
```

As you can see, each database shard has the `NAME` attribute, which indicates the shard name corresponding to the database shard. Each shard name uniquely corresponds to one database shard name. For example, the shard name `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0003_RDS` corresponds to the database shard name `drds_test_vtla_0003`. Therefore, after obtaining the shard name, you can use the hint to specify the corresponding database shard to run the SQL statement.

- Specify database shard 0 to run an SQL statement:

```
SELECT /*TDDL:node('DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS')*/ * FROM table_name;
```

- Specify multiple database shards to run an SQL statement:

```
SELECT /*TDDL:node('DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS','DRDS_TEST_1473471355140LR
PRDRDS_TEST_VTLA_0006_RDS')*/ * FROM table_name;
```

  This SQL statement will be executed in the database shards whose shard names are `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS` and `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0006_RDS`.

- View the execution plan of an SQL statement in database shard 0:

```
/*TDDL:node('DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS')*/ EXPLAIN SELECT * FROM table_na
me;
```

13.1.8.5. Scan all or some of database shards and table shards

In addition to routing an SQL statement to one or more database shards for execution, provides the `SCAN HINT` to scan all or some of database shards and table shards. With the `SCAN HINT`, you can route an SQL statement to each database shard at a time. For example, you can view all the table shards in a specified database shard or view the data volume of each physical table of a specified logical table.

> **Note** This topic is applicable to 5.3 and later. For earlier versions, see Scan all database shards and table shards.

With the SCAN HINT, you can specify the following SQL execution manners:

- Run an SQL statement in all table shards in all database shards.
- Run an SQL statement in all table shards in a specified database shard.
- Run an SQL statement in the specified table shard in the specified database shard by calculating the name of the physical table based on conditions.
- Run an SQL statement in the specified table shard in the specified database shard by explicitly specifying the name of the physical table.

The `SCAN HINT` can be used in data manipulation language (DML) statements, data definition language (DDL) statements, and some data access language (DAL) statements.

> **Note**
>
> - hints can be in the formats of `/*+TDDL:hint_command*/` and `/! +TDDL:hint_command*/`.
> - In the MySQL command-line client, you may need to execute an SQL statement that contains a hint in the format of `/*+TDDL:hint_command*/`. In this case, add the `-c` parameter to the logon command. Otherwise, the client deletes the hint before it sends the SQL statement to the server for execution because the hint is in the format of a MySQL comment. In this case, the hint fails to take effect. For more information, see MySQL Client Options.

Syntax

```
# SCAN HINT
# Route an SQL statement to all table shards in all database shards.
SCAN()
# Route an SQL statement to all table shards in a specified database shard.
SCAN(NODE="node_list")               # Specify the database shard.
# Route an SQL statement to the specified table shard in the specified database shard by calculating
the name of the physical table based on conditions.
SCAN(
  [TABLE=]"table_name_list"          # Specify the name of the logical table.
  , CONDITION="condition_string"     # Calculate the names of physical databases based on the content
of TABLE and CONDITION.
  [, NODE="node_list"] )             # Filter the results obtained based on the content of CONDITION,
to retain only the results of the specified physical database.
# Route an SQL statement to the specified table shard in the specified database shard by explicitly s
pecifying the name of the physical table.
SCAN(
  [TABLE=]"table_name_list"          # Specify the name of the logical table.
  , REAL_TABLE=("table_name_list")   # Specify the name of the physical table. The same physical tabl
e names are applied to all physical databases.
  [, NODE="node_list"] )             # Filter the results obtained based on the content of CONDITION,
to retain only the results of the specified physical database.
# Specify physical table names or logical table names.
table_name_list:
    table_name [, table_name]...
# Specify physical databases by using GROUP_KEY and GROUP_INDEX, which can be obtained by running the
SHOW NODE statement.
node_list:
    {group_key | group_index} [, {group_key | group_index}]...
# Run an SQL WHERE statement. When using this syntax, you must specify conditions for each table, for
example, t1.id = 2 and t2.id = 2.
condition_string:
    where_condition
```

Examples

- Run the following SQL statement in all table shards in all database shards:

  ```
  SELECT /*+TDDL:scan()*/ COUNT(1) FROM t1
  ```

  After this statement is executed, the SQL statement is routed to all the physical tables corresponding to the logical table `t1`, and the result sets are merged and returned.

- Run the following SQL statement in all table shards in specified database shards:

  ```
  SELECT /*+TDDL:scan(node='0,1,2')*/ COUNT(1) FROM t1
  ```

  After this statement is executed, all physical tables corresponding to the logical table `t1` in database shards 0000, 0001, and 0002 are calculated, the SQL statement is routed to the physical tables, and the result sets are merged and returned.

- Run the following SQL statement in specified table shards based on conditions:

  ```
  SELECT /*+TDDL:scan('t1', condition='t1.id = 2')*/ COUNT(1) FROM t1
  ```

  After this statement is executed, all physical tables that correspond to the logical table `t1` and meet the conditions are calculated, the SQL statement is routed to the physical tables, and the result sets are merged and returned.

- Run the following SQL JOIN statement in the specified table shards based on conditions:

```
SELECT /*+TDDL:scan('t1, t2', condition='t1.id = 2 and t2.id = 2')*/ * FROM t1 a JOIN t2 b ON a.id
= b.id WHERE b.name = "test"
```

After this statement is executed, all physical tables that correspond to the logical tables `t1` and `t2` and meet the conditions are calculated, the SQL statement is routed to the physical tables, and the result sets are merged and returned.

> **Notice** Before using this custom hint, you must ensure that the logical tables t1 and t2 are partitioned into the same number of database shards and the same number of table shards. Otherwise, the database shards calculated by the instance based on the conditions are different, and an error will be returned.

- Run the following SQL statement in the specified table shards in database shards by explicitly specifying the names of the physical tables:

```
SELECT /*+TDDL:scan('t1', real_table=("t1_00", "t1_01"))*/ COUNT(1) FROM t1
```

After this statement is executed, the SQL statement is routed to the table shards `t1_00``t1_01` in all database shards, and the result sets are merged and returned.

- Run the following SQL JOIN statement in the specified table shards in database shards by explicitly specifying the names of the physical tables:

```
SELECT /*+TDDL:scan('t1, t2', real_table=("t1_00,t2_00", "t1_01,t2_01"))*/ * FROM t1 a JOIN t2 b O
N a.id = b.id WHERE b.name = "test";
```

After this statement is executed, the SQL statement is routed to the table shards `t1_00`, `t2_00`, `t1_01`, and `t2_01` in all database shards, and the result sets are merged and returned.

13.1.8.6. INDEX HINT

- supports global secondary indexes. The INDEX hint allows you to obtain query results from a specified GSI.
- The INDEX hint takes effect only for SQL SELECT statements.

> **Note** This custom hint is applicable to only MySQL 5.7 and later and 5.4.1 and later.

Syntax

```
# FORCE INDEX
tbl_name [[AS] alias] [index_hint]
index_hint:
    FORCE INDEX({index_name})
# INDEX()
/*+TDDL:
    INDEX({table_name | table_alias}, {index_name})
*/
```

INDEX hint can be used in two ways:

- `FORCE INDEX()`: This syntax is the same as that of MySQL FORCE INDEX.
- `INDEX()`: In this syntax, a global secondary index is specified using a table name (or alias) and an index name. This hint does not take effect in the following cases:
  - The query does not contain the specified table name or alias.
  - The specified global secondary index is not in the specified table.

> **Note**
> - hints can be in the formats of `/*+TDDL:hint_command*/` and `/! +TDDL:hint_command*/`.
> - In the MySQL command-line client, you may need to execute an SQL statement that contains a hint in the format of `/*+TDDL:hint_command*/`. In this case, add the `-c` parameter to the logon command. Otherwise, the client deletes the hint before it sends the SQL statement to the server for execution because the hint is in the format of a MySQL comment. In this case, the hint fails to take effect. For more information, see MySQL Client Options.

Examples

```
CREATE TABLE t_order (
 `id` bigint(11) NOT NULL AUTO_INCREMENT,
 `order_id` varchar(20) DEFAULT NULL,
 `buyer_id` varchar(20) DEFAULT NULL,
 `seller_id` varchar(20) DEFAULT NULL,
 `order_snapshot` longtext DEFAULT NULL,
 `order_detail` longtext DEFAULT NULL,
 PRIMARY KEY (`id`),
 GLOBAL INDEX `g_i_seller`(`seller_id`) dbpartition by hash(`seller_id`),
 UNIQUE GLOBAL INDEX `g_i_buyer` (`buyer_id`) COVERING(`seller_id`, `order_snapshot`)
  dbpartition by hash(`buyer_id`) tbpartition by hash(`buyer_id`) tbpartitions 3
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by hash(`order_id`);
```

Specify the global secondary index `g_i_seller` by using FORCE INDEX in the FROM clause:

```
 SELECT a.*, b.order_id
   FROM t_seller a
     JOIN t_order b FORCE INDEX(g_i_seller) ON a.seller_id = b.seller_id
   WHERE a.seller_nick="abc";
```

Specify the global secondary index `g_i_buyer` by using INDEX+table alias:

```
/*+TDDL:index(a, g_i_buyer)*/ SELECT * FROM t_order a WHERE a.buyer_id = 123
```

13.1.9. PolarDB-X 5.2 hints

13.1.9.1. Introduction to hints

As a supplement to the SQL syntax, hints play a critical role in relational databases. They allow you to affect execution plans of SQL statements by using relevant syntax, to specially optimize the SQL statements.

Overview of PolarDB-X hints

provides special hint syntax.

For example, if you know the target data is stored in table shards in certain database shards and you need to route the SQL statement directly to the database shards for execution, you can use custom hints provided by .

```
/! TDDL:NODE IN('node_name', ...) */SELECT * FROM table_name;
```

In the preceding SQL statement, the part between `/!` and `*/`, namely, `TDDL:node in('node_name', ...)`, is a hint. The hint specifies the ApsaraDB RDS for MySQL database shard where the SQL statement is to be executed.

> **Note**
> - hints can be in the formats of `/*+TDDL:hint_command*/` and `/! +TDDL:hint_command*/`.
> - In the MySQL command-line client, you may need to execute an SQL statement that contains a hint in the format of `/*+TDDL:hint_command*/`. In this case, add the `-c` parameter to the logon command. Otherwise, the client deletes the hint before it sends the SQL statement to the server for execution because the hint is in the format of a MySQL comment. In this case, the hint fails to take effect. For more information, see MySQL Client Options.

PolarDB-X hint syntax

Basic syntax:

```
/! TDDL:hint command*/
```

hints are based on MySQL Comment Syntax. Therefore, an SQL statement that contains a hint is located between `/!` and `*/`, and must begin with `TDDL:`. The `hint command` indicates a hint command related to the specific operation. For example, a hint is added to the following SQL statement to display the name of each database shard.

```
/! TDDL:SCAN*/SHOW TABLES;
```

In this SQL statement, `/! TDDL:SCAN*/` is the hint that begins with `TDDL:`, and `SCAN` is a hint command.

13.1.9.2. Read/write splitting

provides transparent read/write splitting at the application layer. Data synchronization between primary and read-only ApsaraDB RDS for MySQL instances has a delay of several milliseconds. If you need to read changed data immediately after the primary ApsaraDB RDS for MySQL instance is changed, you must ensure that the SQL statement for reading data is routed to the primary ApsaraDB RDS for MySQL instance. To meet this demand, provides custom hints for read/write splitting, to route SQL statements to a specified primary or read-only ApsaraDB RDS for MySQL instance.

Syntax

```
/! TDDL:MASTER|SLAVE*/
```

With this custom hint, you can specify whether to run an SQL statement on a primary or read-only ApsaraDB RDS for MySQL instance. With the custom hint `/!TDDL:SLAVE*/`, if a primary ApsaraDB RDS for MySQL instance is configured with multiple read-only ApsaraDB RDS for MySQL instances, the instance randomly selects a read-only ApsaraDB RDS for MySQL instance based on its weight, to run the SQL statement.

> **Note**
> - hints can be in the formats of `/*+TDDL:hint_command*/` and `/! +TDDL:hint_command*/`.
> - In the MySQL command-line client, you may need to execute an SQL statement that contains a hint in the format of `/*+TDDL:hint_command*/`. In this case, add the `-c` parameter to the logon command. Otherwise, the client deletes the hint before it sends the SQL statement to the server for execution because the hint is in the format of a MySQL comment. In this case, the hint fails to take effect. For more information, see MySQL Client Options.

Examples

- Specify a primary ApsaraDB RDS for MySQL instance to run an SQL statement:

```
/! TDDL:MASTER*/SELECT * FROM table_name;
```

After the custom hint `/! TDDL:MASTER*/` is added at the beginning of the SQL statement, this SQL statement is routed to the primary ApsaraDB RDS for MySQL instance for execution.

- Specify a read-only ApsaraDB RDS for MySQL instance to run an SQL statement:

```
/! TDDL:SLAVE*/SELECT * FROM table_name;
```

After the custom hint `/! TDDL:SLAVE*/` is added at the beginning of the SQL statement, this SQL statement is randomly routed to a read-only ApsaraDB RDS for MySQL instance based on the allocated weight.

Considerations

- The custom hints for read/write splitting are only applicable to read SQL statements for non-transactional data. SQL statements for transactional data and write SQL statements are still routed to the primary ApsaraDB RDS for MySQL instance.
- When you use the `/*+TDDL:slave()*/` hint, the instance routes the SQL statement randomly to a read-only ApsaraDB RDS for MySQL instance based on the allocated weight. If no read-only ApsaraDB RDS for MySQL instance is available, no error is reported. Instead, the primary ApsaraDB RDS for MySQL instance is selected to execute the SQL statement.

13.1.9.3. Prevent the delay from a read-only ApsaraDB RDS for MySQL instance

Normally, if you have configured a read-only ApsaraDB for RDS instance for the primary ApsaraDB RDS for MySQL instance of a logical database in a instance and set read traffic for both the primary and read-only ApsaraDB RDS for MySQL instances, routes SQL statements to the primary and read-only ApsaraDB RDS for MySQL instances based on the read/write ratio. However, if asynchronous data replication between the primary and read-only ApsaraDB RDS for MySQL instances has a high delay, an error is reported or error results are returned when PolarDB-X routes the SQL statements to the read-only ApsaraDB RDS for MySQL instance.

To address this issue, the PolarDB-X instance provides a custom hint to cut off the delay of the read-only instance. Specifically, based on the maximum delay of primary/secondary replication, PolarDB-X determines whether to route the SQL statement to the primary or the read-only ApsaraDB RDS for MySQL instance.

Syntax

```
/! TDDL:SQL_DELAY_CUTOFF=time*/
```

With this custom hint, you can specify the value of `SQL_DELAY_CUTOFF`. When the value of SQL_DELAY (primary/secondary replication delay of ApsaraDB RDS for MySQL) for the read-only ApsaraDB RDS for MySQL instance reaches or exceeds the value of `time` (which is measured in seconds), the SQL statement is routed to the primary ApsaraDB RDS for MySQL instance.

> **Note**
>
> - hints can be in the formats of `/*+TDDL:hint_command*/` and `/! +TDDL:hint_command*/`.
> - In the MySQL command-line client, you may need to execute an SQL statement that contains a hint in the format of `/*+TDDL:hint_command*/`. In this case, add the `-c` parameter to the logon command. Otherwise, the client deletes the hint before it sends the SQL statement to the server for execution because the hint is in the format of a MySQL comment. In this case, the hint fails to take effect. For more information, see MySQL Client Options.

Examples

- Set the primary/secondary replication delay to 5 seconds:

```
/! TDDL:SQL_DELAY_CUTOFF=5*/SELECT * FROM table_name;
```

  In this SQL statement, the value of `SQL_DELAY_CUTOFF` is set to 5. Therefore, when the value of `SQL_DELAY` for the read-only ApsaraDB RDS for MySQL instance reaches or exceeds 5 seconds, the SQL statement is routed to the primary ApsaraDB RDS for MySQL instance.

- Use the custom hint for delay cutoff with other custom hints:

```
/! TDDL:SLAVE AND SQL_DELAY_CUTOFF=5*/SELECT * FROM table_name;
```

  The custom hint for cutting off the delay of the read-only ApsaraDB RDS for MySQL instance can be used with other hints. By default, the SQL query request is routed to a read-only ApsaraDB RDS for MySQL instance. However, when the primary/secondary replication delay reaches or exceeds 5 seconds, the SQL query request is routed to the primary ApsaraDB RDS for MySQL instance.

13.1.9.4. Specify a timeout period for an SQL statement

In , the SQL statements for instances and ApsaraDB RDS for MySQL instances are timed out after 900 seconds (which can be adjusted) by default. However, for some slow SQL statements, the execution duration may exceed 900 seconds. For these slow SQL statements, provides a custom hint to adjust their timeout periods. You can use this custom hint to adjust the SQL execution duration as needed.

Syntax

The syntax of the hint for specifying a timeout period for an SQL statement is as follows:

```
/! TDDL:SOCKET_TIMEOUT=time*/
```

The `SOCKET_TIMEOUT` parameter is measured in milliseconds. With this custom hint, you can adjust the timeout period for the SQL statement based on business requirements.

> **Note**
> - hints can be in the formats of `/*+TDDL:hint_command*/` and `/! +TDDL:hint_command*/`.
> - In the MySQL command-line client, you may need to execute an SQL statement that contains a hint in the format of `/*+TDDL:hint_command*/`. In this case, add the `-c` parameter to the logon command. Otherwise, the client deletes the hint before it sends the SQL statement to the server for execution because the hint is in the format of a MySQL comment. In this case, the hint fails to take effect. For more information, see MySQL Client Options.

Examples

Set the timeout period of an SQL statement to 40 seconds:

```
/! TDDL:SOCKET_TIMEOUT=40000*/SELECT * FROM t_item;
```

> **Note** A longer timeout period causes database resources to be occupied for a longer period of time. If excessive SQL statements are executed over a long time within the same period, a large number of database resources may be consumed. This will make users unable to use PolarDB-X properly. In this case, we need to use this custom hint to optimize the SQL statements that take a long time to execute.

13.1.9.5. Specify a database shard to run an SQL statement

When running SQL commands in a instance, you may find that some SQL statements are not supported by the instance. In this case, you can use the custom hint provided by to route the SQL statements to one or more database shards for execution. In addition, if you need to query the data in a specified database shard or the data in a specified table shard, you can use the custom hint to directly route the SQL statement to the database shard for execution.

Syntax

This custom hint allows you to specify a database shard by using a shard name or the value of the database shard key, to run an SQL statement in the database shard. A shard name uniquely identifies a database shard in a instance. You can run the `SHOW NODE` command to obtain the shard name.

> **Note** If the hint for specifying a database shard is used in an INSERT statement that contains a sequence for the target table, the sequence will not take effect. For more information, see Limits and precautions for sequences.

- Specify a database shard by using a shard name, to run an SQL statement

  This custom hint allows you to specify one or more database shards to run an SQL statement.

  - Specify one database shard to run an SQL statement:

    ```
    /! TDDL:NODE='node_name'*/
    ```

    Specifically, `node_name` indicates the shard name. This hint enables you to route the SQL statement to the database shard specified by `node_name`.

  - Specify multiple database shards to run an SQL statement:

    ```
    /! TDDL:NODE IN ('node_name'[,'node_name1','node_name2'])*/
    ```

    The `IN` keyword is used to specify multiple shard names. This custom hint allows you to route the SQL statement to multiple database shards. Separate multiple shard names with commas (,).

    > **Note** When this custom hint is used, the instance directly routes the SQL statement to the specified database shards for execution. Therefore, the specified shard names in the SQL statement must correspond to existing database shards.

- Specify a database shard by using the value of the database shard key, to run an SQL statement

  ```
  /! TDDL:table_name.partition_key=value [and table_name1.partition_key=value1]*/
  ```

  In this hint, `table_name` indicates the name of a logical table, and this table is a partitioned table. In addition, `partition_key` indicates a shard key, and `value` indicates the value specified for the shard key. In this custom hint, you can use the `and` keyword to specify the shard keys of multiple partitioned tables. When this hint is used, the instance calculates the database shards and table shards where the SQL statement is to be executed, and routes the SQL statement to the corresponding database shards.

Note

- hints can be in the formats of `/*+TDDL:hint_command*/` and `/! +TDDL:hint_command*/`.
- In the MySQL command-line client, you may need to execute an SQL statement that contains a hint in the format of `/*+TDDL:hint_command*/`. In this case, add the `-c` parameter to the logon command. Otherwise, the client deletes the hint before it sends the SQL statement to the server for execution because the hint is in the format of a MySQL comment. In this case, the hint fails to take effect. For more information, see MySQL Client Options.

Examples

The following shows the responses of the `SHOW NODE` statement for a logical database named `drds_test` in a instance.

```
mysql> SHOW NODE\G
*************************** 1. row ******************
                ID: 0
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS
  MASTER_READ_COUNT: 212
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 2. row ******************
                ID: 1
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0001_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 3. row ******************
                ID: 2
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0002_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 4. row ******************
                ID: 3
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0003_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 5. row ******************
                ID: 4
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0004_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 6. row ******************
                ID: 5
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0005_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 7. row ******************
                ID: 6
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0006_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
*************************** 8. row ******************
                ID: 7
              NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0007_RDS
  MASTER_READ_COUNT: 29
   SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
 SLAVE_READ_PERCENT: 0%
8 rows in set (0.02 sec)
```

As you can see, each database shard has the `NAME` attribute, which indicates the shard name corresponding to the database shard. Each shard name uniquely corresponds to one database shard name. For example, the shard name `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0003_RDS` corresponds to the database shard name `drds_test_vtla_0003`. Therefore, after obtaining the shard name, you can use the hint to specify the corresponding database shard to run the SQL statement.

- Specify database shard 0 to run an SQL statement:

```
/! TDDL:NODE='DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS'*/SELECT * FROM table_name;
```

- Specify multiple database shards to run an SQL statement:

```
/! TDDL:NODE  IN('DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS','DRDS_TEST_1473471355140LRPR
DRDS_TEST_VTLA_0006_RDS')*/SELECT * FROM table_name;
```

  This SQL statement will be executed in the database shards whose shard names are `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS` and `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0006_RDS`.

- View the execution plan of an SQL statement in a specified database shard:

```
/! TDDL:NODE='DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS'*/EXPLAIN SELECT * FROM table_nam
e;
```

  After this SQL statement is executed, the execution plan of the `SELECT` statement in the database shard corresponding to the shard name `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS` will be returned.

- Specify a database shard by using the value of the database shard key, to run an SQL statement:

  does not support subqueries in the `SET` clause of an `UPDATE` statement, because a shard key must be specified for `UPDATE` statements in . To address this issue, provides a custom hint to route the statement to a database shard for execution.

  For example, the following shows the CREATE TABLE statement for creating two logical tables t1 and t2, which are partitioned into table shards in database shards:

```
CREATE TABLE `t1` (
    `id` bigint(20) NOT NULL,
    `name` varchar(20) NOT NULL,
    `val` varchar(20) DEFAULT NULL,
    PRIMARY KEY (`id`)
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by hash(`id`) tbpartition by hash(`name`) tbpar
titions 3
  CREATE TABLE `t2` (
    `id` bigint(20) NOT NULL,
    `name` varchar(20) NOT NULL,
    `val` varchar(20) DEFAULT NULL,
    PRIMARY KEY (`id`)
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by hash(`id`) tbpartition by hash(`name`) tbpar
titions 3
```

  The following SQL statement is to be executed for the two tables:

```
UPDATE t1 SET val=(SELECT val FROM t2 WHERE id=1) WHERE id=1;
```

  If this statement is directly executed in a instance, an error will be returned indicating that this statement is not supported. In this case, you can add the hint to this SQL statement before submitting it to the instance for execution. The SQL statements are as follows:

```
/! TDDL:t1.id=1 and t2.id=1*/UPDATE t1 SET val=(SELECT val FROM t2 WHERE id=1) WHERE id=1;
```

This statement will be routed to database shards of `t1`, with the `id` of the database shards being 1. You can run the following `EXPLAIN` command to view the execution plan of this SQL statement:

```
  mysql> explain /! TDDL:t1.id=1 and t2.id=1*/UPDATE t1 SET val=(SELECT val FROM t2 WHERE id=1) WH
ERE id=1\G
  *************************** 1. row ***************************
  GROUP_NAME: TEST_DRDS_1485327111630IXLWTEST_DRDS_IGHF_0001_RDS
         SQL: UPDATE `t1_2` AS `t1` SET `val` = (SELECT val FROM `t2_2` AS `t2` WHERE `id` = 1) WH
ERE `id` = 1
      PARAMS: {}
  *************************** 2. row ***************************
  GROUP_NAME: TEST_DRDS_1485327111630IXLWTEST_DRDS_IGHF_0001_RDS
         SQL: UPDATE `t1_1` AS `t1` SET `val` = (SELECT val FROM `t2_1` AS `t2` WHERE `id` = 1) WH
ERE `id` = 1
      PARAMS: {}
  *************************** 3. row ***************************
  GROUP_NAME: TEST_DRDS_1485327111630IXLWTEST_DRDS_IGHF_0001_RDS
         SQL: UPDATE `t1_0` AS `t1` SET `val` = (SELECT val FROM `t2_0` AS `t2` WHERE `id` = 1) WH
ERE `id` = 1
      PARAMS: {}
  3 rows in set (0.00 sec)
```

According to the result set of the `EXPLAIN` command, the SQL statement is rewritten into three statements, which are then routed to the database shards for execution. You can further specify a table shard by using the value of the table shard key, to narrow the execution scope of the SQL statement to a specified table shard.

```
  mysql> explain /! TDDL:t1.id=1 and t2.id=1 and t1.name='1'*/UPDATE t1 SET val=(SELECT val FROM t
2 WHERE id=1) WHERE id=1\G
  *************************** 1. row ***************************
  GROUP_NAME: TEST_DRDS_1485327111630IXLWTEST_DRDS_IGHF_0001_RDS
         SQL: UPDATE `t1_1` AS `t1` SET `val` = (SELECT val FROM `t2_1` AS `t2` WHERE `id` = 1) WH
ERE `id` = 1
      PARAMS: {}
  1 row in set (0.00 sec)
```

Note Before using this custom hint, you must ensure that the logical tables t1 and t2 are partitioned into the same number of database shards and the same number of table shards. Otherwise, the database shards calculated by the instance based on the conditions are different, and an error will be returned.

13.1.9.6. Scan all database shards and table shards

In addition to routing an SQL statement to one or more database shards for execution, provides a custom hint to allow you to scan all database shards and table shards. With this custom hint, you can route an SQL statement to each database shard at a time. For example, you can use this custom hint to view all the table shards in a specified database shard. In addition, you can use this custom hint to view the data volume of table shards in each database shard corresponding to a specified logical table.

Syntax

With this hint, you can route an SQL statement to all database shards for execution and route an SQL statement to all database shards to perform an operation on a specified logical table.

- Route an SQL statement to all database shards for execution:

```
/! TDDL:SCAN*/
```

- Perform an operation on a specified logical table:

```
/! TDDL:SCAN='table_name'*/
```

The `table_name` parameter indicates the name of a logical table in the logical database of a instance. This custom hint is provided for table shards in database shards. Ensure that the value of `table_name` is the name of a table shard in database shards.

> **Note**
> - hints can be in the formats of `/*+TDDL:hint_command*/` and `/! +TDDL:hint_command*/`.
> - In the MySQL command-line client, you may need to execute an SQL statement that contains a hint in the format of `/*+TDDL:hint_command*/`. In this case, add the `-c` parameter to the logon command. Otherwise, the client deletes the hint before it sends the SQL statement to the server for execution because the hint is in the format of a MySQL comment. In this case, the hint fails to take effect. For more information, see MySQL Client Options.

Examples

- View the data volume of a specified broadcast table in each database shard:

```
/! TDDL:SCAN*/SELECT COUNT(1) FROM table_name
```

In this SQL statement, `table_name` indicates a broadcast table. This hint causes the PolarDB-X instance to route the SQL statement to each database shard for execution. Therefore, the result sets include the total data volume of the broadcast table `table_name` in all database shards. This statement allows you to conveniently check whether the data of a broadcast table is normal.

- Scan a single-database non-partition logical table:

```
/! TDDL:SCAN*/SELECT COUNT(1) FROM table_name
```

This hint causes the PolarDB-X instance to route the SQL `select count(1) from table_name` statement to each database shard for execution. The `table_name` parameter indicates a logical table in a logical database of a instance. Before using this hint, ensure that each database shard contains the table shard `table_name`. In other words, the table shard `table_name` is a logical table that is only partitioned into database shards, but not partitioned into table shards. Otherwise, an error that indicates that the table is not found will be returned.

- Scan a partitioned logical table in database shards:

```
/! TDDL:SCAN='table_name'*/SELECT COUNT(1) FROM table_name
```

When executing this statement, the instance first calculates all the database shards and table shards corresponding to the logical table `table_name`, and then generates a COUNT clause for each table shard in each database shard.

- View the execution plans of all database shards:

```
/! TDDL:SCAN='table_name'*/EXPLAIN SELECT * FROM table_name;
```

13.1.10. Distributed transactions

13.1.10.1. Distributed transactions based on MySQL 5.7

> **Note**
> - When you use MySQL 5.7 or later and 5.3.4 or later, XA distributed transactions are automatically enabled. The user experience of the XA distributed transactions is the same as that of single-database transactions in MySQL. No special commands are required to enable XA distributed transactions.
> - When you use MySQL and a PolarDB-X instance in other versions, see Distributed transactions based on MySQL 5.6.

How it works

When you use MySQL 5.7 or later, the instance processes distributed transactions based on the XA protocol by default.

Use method

The user experience of distributed transactions in a instance is the same as that of single-database transactions in MySQL, for example, in terms of the following commands:

- `SET AUTOCOMMIT=0` : Start a transaction.
- `COMMIT` : Commit the current transaction.
- `ROLLBACK` : Roll back the current transaction.

If the SQL statement in a transaction involves only a single shard, the instance routes the transaction directly to the ApsaraDB RDS for MySQL instance as a single-database transaction. If the SQL statement in the transaction is to modify the data of multiple shards, the instance automatically upgrades the current transaction to a distributed transaction.

13.1.10.2. Distributed transactions based on MySQL 5.6

How it works

The XA protocol for MySQL 5.6 is not mature. Therefore, the instance independently implements two-phase commit (2PC) transaction policies for distributed transactions. When you use MySQL 5.7 or later, we recommend that you use XA transaction policies.

> **Note** The distributed transactions described in this topic are intended for users who use MySQL 5.6 or earlier than 5.3.4. When you use MySQL 5.7 or later and a instance in 5.3.4 or later, see Distributed transactions based on MySQL 5.7.

Use method

If a transaction involves multiple database shards, you must declare the current transaction as a distributed transaction. If a transaction involves only a single database shard, you do not need to enable distributed transactions, but can process the transaction as a single-database transaction in MySQL. No additional operations are required.

To enable distributed transactions, do as follows:

After transactions are enabled, run `SET drds_transaction_policy = '...'` .

To enable 2PC transactions in the MySQL command-line client, run the following statements:

```
SET AUTOCOMMIT=0;
SET drds_transaction_policy = '2PC'; -- We recommend that you use MySQL 5.6 to run this command.
.... -- Here, you can run your business SQL statement.
COMMIT; -- You can alternatively write ROLLBACK.
```

To enable 2PC transactions by using the Java database connectivity (JDBC) API, write the code as follows:

```
conn.setAutoCommit(false);
try (Statement stmt = conn.createStatement()) {
    stmt.execute("SET drds_transaction_policy = '2PC'");
}
// ... Here, you can run your business SQL statement.
conn.commit(); // You can alternatively write rollback().
```

FAQ

Q: How can I use distributed transactions in the Spring framework?

A: If you enable transactions by using the Spring `@Transactional` annotation, you can enable distributed transactions by extending the transaction manager.

Sample code:

```
import org.springframework.jdbc.datasource.DataSourceTransactionManager;
import org.springframework.transaction.TransactionDefinition;
import javax.sql.DataSource;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Statement;
public class DrdsTransactionManager extends DataSourceTransactionManager {
    public DrdsTransactionManager(DataSource dataSource) {
        super(dataSource);
    }
    @Override
    protected void prepareTransactionalConnection(Connection con, TransactionDefinition definition) t
hrows SQLException {
        try (Statement stmt = con.createStatement()) {
            stmt.executeUpdate("SET drds_transaction_policy = '2PC'"); // A 2PC transaction is used a
s an example.
        }
    }
}
```

After that, instantiate the preceding class in the Spring configuration. For example, you can write the code as follows:

```
<bean id="drdsTransactionManager" class="my.app.DrdsTransactionManager">
     <property name="dataSource" ref="yourDataSource" />
</bean>
```

To enable distributed transactions for a class, you can add the `@Transactional("drdsTransactionManager")` annotation.

13.1.11. DDL operations

13.1.11.1. DDL statements

The data definition language (DDL) statement CREATE TABLE in a Distributed Relational Database Service (DRDS) instance is similar to that in a MySQL database, and is extended based on the syntax in a MySQL database. To create a table shard in a DRDS instance, you must specify the table sharding manner and the database sharding manner in the `drds_partition_options` parameter. The valid values include `DBPARTITION BY` , `TBPARTITION BY` , `TBPARTITIONS` , and `BROADCAST` .

Currently, you can run a DDL statement in the following ways:

- Run the DDL statement through the MySQL command-line client, for example, by using MySQL command lines, Navicat, or MySQL Workbench.
- Connect to the specified DRDS instance by using program code and then call the DDL statement for execution.

For the syntax of the CREATE TABLE statement in a MySQL database, see MySQL CREATE TABLE Statement.

13.1.11.2. CREATE TABLE statement

13.1.11.2.1. Overview

This topic describes the syntax, clauses, parameters, and basic methods for creating a table by using a data definition language (DDL) statement.

> **Note** instances do not allow you to directly create a database by using a DDL statement. To create a database, you can Log on to the PolarDB-X console. For the information about how to create a database, see Create a database.

Syntax

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    (create_definition,...)
    [table_options]
    [drds_partition_options]
    [partition_options]
drds_partition_options:
    DBPARTITION BY
        HASH([column])
        [TBPARTITION BY
            { HASH(column)
            | {MM|DD|WEEK|MMDD}(column)}
          [TBPARTITIONS num]
        ]
```

Clauses and parameters for database and table sharding

- `DBPARTITION BY hash(partition_key)` : This parameter specifies the shard key and the sharding algorithm for database sharding. Database sharding by time is not supported.
- `TBPARTITION BY { HASH(column) | {MM|DD|WEEK|MMDD}(column)` : (Optional) This parameter specifies the method of mapping data to a physical table. The value is the same as that of DBPARTITION BY by default.
- `TBPARTITIONS num` : (Optional) This parameter specifies the number of physical tables to be created in each database shard. The default value is 1. If no table sharding is required, you do not need to specify this parameter.

13.1.11.2.2. Create a single-database non-partition table

This topic describes how to create a single-database non-partition table.

Create a single-database non-partition table

```
CREATE TABLE single_tbl(
  id int,
  name varchar(30),
  primary key(id)
);
```

According to the node topology of the logical table, you can see that a single-database non-partition logical table is created in database 0.

```
mysql> show topology from single_tbl;
+------+--------------------------------------------------------------------+------------+
| ID   | GROUP_NAME                                                         | TABLE_NAME |
+------+--------------------------------------------------------------------+------------+
|    0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | single_tbl |
+------+--------------------------------------------------------------------+------------+
1 row in set (0.01 sec)
```

Specify parameters

You can also specify the `select_statement` parameter when creating a a single-database non-partition table. If you need to create table shards, you cannot specify this parameter.

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    [(create_definition,...)]
    [table_options]
    [partition_options]
    select_statement
```

For example, you can run the following statement to create a single-database non-partition table single_tbl2 to store the data from the single_tbl table. In this case, no sharding is required.

```
CREATE TABLE single_tbl2(
  id int,
  name varchar(30),
  primary key(id)
) select * from single_tbl;
```

13.1.11.2.3. Create a non-partition table in database shards

This topic describes how to create a non-partition table in database shards.

Assume that eight database shards have been created. You can run the following command to create a non-partition table in the database shards by calculating the hash function based on the userId shard key.

```
CREATE TABLE multi_db_single_tbl(
  id int,
  name varchar(30),
  primary key(id)
) dbpartition by hash(id);
```

According to the node topology of the logical table, you can see that a table shard is created in each database shard. In other words, the table is only distributed to database shards.

```
mysql> show topology from multi_db_single_tbl;
+------+-------------------------------------------------------------------+---------------------+
| ID   | GROUP_NAME                                                        | TABLE_NAME          |
+------+-------------------------------------------------------------------+---------------------+
|    0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | multi_db_single_tbl |
|    1 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | multi_db_single_tbl |
|    2 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0002_RDS | multi_db_single_tbl |
|    3 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0003_RDS | multi_db_single_tbl |
|    4 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0004_RDS | multi_db_single_tbl |
|    5 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS | multi_db_single_tbl |
|    6 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0006_RDS | multi_db_single_tbl |
|    7 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | multi_db_single_tbl |
+------+-------------------------------------------------------------------+---------------------+
8 rows in set (0.01 sec)
```

13.1.11.2.4. Create table shards in database shards

This topic describes how to create table shards in database shards in different sharding manners.

- Use HASH for sharding
- Use RANGE_HASH for sharding
- Use date functions for sharding

In the following examples, it is assumed that eight database shards have been created.

Use HASH for sharding

Create a table that is split into table shards in database shards, with each database shard containing three physical tables. The database sharding process is calculating the hash function by using id as the shard key, and the table sharding process is calculating the hash function by using bid as the shard key. Specifically, a hash operation is performed on the data of the table based on the id column, to distribute the data to multiple database shards. Then, a hash operation is performed on the data in each database shard based on the bid column, to distribute the data to the three physical tables.

```
CREATE TABLE multi_db_multi_tbl(
  id int auto_increment,
  bid int,
  name varchar(30),
  primary key(id)
) dbpartition by hash(id) tbpartition by hash(bid) tbpartitions 3;
```

According to the node topology of the logical table, you can see that three table shards are created in each database shard.

```
mysql> show topology from multi_db_multi_tbl;
+------+-------------------------------------------------------------------+-----------------------+
| ID   | GROUP_NAME                                                        | TABLE_NAME            |
+------+-------------------------------------------------------------------+-----------------------+
|    0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | multi_db_multi_tbl_00 |
|    1 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | multi_db_multi_tbl_01 |
|    2 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | multi_db_multi_tbl_02 |
|    3 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | multi_db_multi_tbl_03 |
|    4 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | multi_db_multi_tbl_04 |
|    5 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | multi_db_multi_tbl_05 |
|    6 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0002_RDS | multi_db_multi_tbl_06 |
|    7 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0002_RDS | multi_db_multi_tbl_07 |
|    8 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0002_RDS | multi_db_multi_tbl_08 |
|    9 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0003_RDS | multi_db_multi_tbl_09 |
|   10 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0003_RDS | multi_db_multi_tbl_10 |
|   11 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0003_RDS | multi_db_multi_tbl_11 |
|   12 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0004_RDS | multi_db_multi_tbl_12 |
|   13 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0004_RDS | multi_db_multi_tbl_13 |
|   14 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0004_RDS | multi_db_multi_tbl_14 |
|   15 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS | multi_db_multi_tbl_15 |
|   16 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS | multi_db_multi_tbl_16 |
|   17 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS | multi_db_multi_tbl_17 |
|   18 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0006_RDS | multi_db_multi_tbl_18 |
|   19 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0006_RDS | multi_db_multi_tbl_19 |
|   20 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0006_RDS | multi_db_multi_tbl_20 |
|   21 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | multi_db_multi_tbl_21 |
|   22 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | multi_db_multi_tbl_22 |
|   23 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | multi_db_multi_tbl_23 |
+------+-------------------------------------------------------------------+-----------------------+
24 rows in set (0.01 sec)
```

According to the sharding rule of the logical table, you can see that both database sharding and table sharding are running the hash function, except that the database shard key is id and the table shard key is bid.

```
mysql> show rule from multi_db_multi_tbl;
+------+--------------------+-----------+------------------+---------------------+-------------------
-+------------------+---------------------+--------------------+
| ID   | TABLE_NAME         | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT
| TB_PARTITION_KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+------+--------------------+-----------+------------------+---------------------+-------------------
-+------------------+---------------------+--------------------+
|    0 | multi_db_multi_tbl |         0 | id               | hash                | 8
| bid              | hash                | 3                  |
+------+--------------------+-----------+------------------+---------------------+-------------------
-+------------------+---------------------+--------------------+
1 row in set (0.01 sec)
```

Use RANGE_HASH for sharding

- Requirements

  The shard key must be a character or a number.

- Routing method

  Calculate a hash value based on the last N digits of any shard key, and then calculate the route by using RANGE_HASH. The number N is the third parameter in the function. For example, during calculation of the RANGE_HASH(COL1, COL2, N) function, COL1 is preferentially selected and then truncated to obtain the last N digits for calculation. If COL1 does not exist, COL2 is selected and truncated for calculation.

- Scenarios

  RANGE_HASH is applicable to scenarios where two shard keys are used for sharding but only the values of one shard is used for SQL query. Assume that a DRDS database is partitioned into eight physical databases. Our customer has the following requirements:

  i. The order table of each service needs to be split into database shards by buyer ID and order ID.

  ii. The query is executed based on either the buyer ID or order ID as the condition.

  In this case, you can run the following DDL statement to create the order table:

```
create table test_order_tb (
           id int,
           seller_id varchar(30) DEFAULT NULL,
           order_id varchar(30) DEFAULT NULL,
           buyer_id varchar(30) DEFAULT NULL,
           create_time datetime DEFAULT NULL,
           primary key(id)
   ) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by RANGE_HASH(buyer_id, order_id, 10) tbpartit
ion by RANGE_HASH(buyer_id, order_id, 10) tbpartitions 3;
```

> **Note**
>
> - Neither of the two shard keys can be modified.
> - Data insertion fails if the two shard keys point to different database shards or table shards.

Use date functions for sharding

In addition to using the hash function as the sharding algorithm, you can also use the date functions MM, DD, WEEK, and MMDD as the table sharding algorithms. For more information, see the following examples.

- Create a table and then split the table into table shards in database shards. The database sharding process is calculating the hash function by using userId as the shard key, and the table sharding process is calculating DAY_OF_WEEK through the WEEK(actionDate) function and then splitting the table into table shards based on the actionDate column, with one week counted as seven days.

  For example, if the value in the actionDate column is 2017-02-27, which is on Monday, the value obtained by calculating the WEEK(actionDate) function is 2. In this case, the record is stored in table shard 2, because 2 % 7 = 2. This table shard is located in a database shard and is named user_log_2. For another example, if the value in the actionDate column is 2017-02-26, which is on Sunday, the value obtained by calculating the WEEK(actionDate) function is 1. In this case, the record is stored in table shard 1, because 1 % 7 = 1. This table shard is located in a database shard and is named user_log_1.

```
CREATE TABLE user_log(
userId int,
name varchar(30),
operation varchar(30),
actionDate DATE
) dbpartition by hash(userId) tbpartition by WEEK(actionDate) tbpartitions 7;
```

  According to the node topology of the logical table, you can see that seven table shards are created in each database shard, because one week is counted as seven days in the function. The responses are very long, and therefore are omitted by using an ellipsis (...).

```
mysql> show topology from user_log;
+------+-------------------------------------------------------------------+------------+
| ID   | GROUP_NAME                                                        | TABLE_NAME |
+------+-------------------------------------------------------------------+------------+
|    0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_0 |
|    1 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_1 |
|    2 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_2 |
|    3 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_3 |
|    4 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_4 |
|    5 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_5 |
|    6 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_6 |
|    7 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_0 |
|    8 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_1 |
|    9 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_2 |
|   10 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_3 |
|   11 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_4 |
|   12 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_5 |
|   13 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_6 |
...
|   49 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_0 |
|   50 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_1 |
|   51 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_2 |
|   52 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_3 |
|   53 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_4 |
|   54 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_5 |
|   55 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_6 |
+------+-------------------------------------------------------------------+------------+
56 rows in set (0.01 sec)
```

According to the sharding rule of the logical table, you can see that the database sharding process is calculating the hash function by using userId as the shard key, and the table sharding process is calculating the WEEK function by using actionDate as the shard key.

```
mysql> show rule from user_log;
+------+------------+-----------+------------------+---------------------+--------------------+---
---------------+---------------------+--------------------+
| ID   | TABLE_NAME | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT | TB
_PARTITION_KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+------+------------+-----------+------------------+---------------------+--------------------+---
---------------+---------------------+--------------------+
|    0 | user_log   |         0 | userId           | hash                | 8                  | ac
tionDate       | week                | 7                  |
+------+------------+-----------+------------------+---------------------+--------------------+---
---------------+---------------------+--------------------+
1 row in set (0.00 sec)
```

According to the specified database and table shard key parameters, you can see the specific physical table in the specific physical database to which the SQL statement is routed.

View the route of the SQL statement

- Create a table that is split into table shards in database shards. The database sharding process is calculating the hash function by using userId as the shard key, and the table sharding process is calculating MONTH_OF_YEAR through the MM(actionDate) function and then splitting the table into table shards based on

the actionDate column, with one year counted as 12 months.

For example, if the value in the actionDate column is 2017-02-27, the value obtained by calculating the MM(actionDate) function is 02. In this case, the record is stored in table shard 02, because 02 % 12 = 02. This table shard is located in a database shard and is named user_log_02. For another example, if the value in the actionDate column is 2016-12-27, the value obtained by calculating the MM(actionDate) function is 12. In this case, the record is stored in table shard 00, because 12 % 12 = 00. This table shard is located in a database shard and is named user_log_00.

```
CREATE TABLE user_log2(
userId int,
name varchar(30),
operation varchar(30),
actionDate DATE
) dbpartition by hash(userId) tbpartition by MM(actionDate) tbpartitions 12;
```

According to the node topology of the logical table, you can see that 12 table shards are created in each database shard, because one year is counted as 12 months in the function. The responses are very long, and therefore are omitted by using an ellipsis (...).

```
mysql> show topology from user_log2;
+------+-------------------------------------------------------------------+--------------+
| ID   | GROUP_NAME                                                        | TABLE_NAME   |
+------+-------------------------------------------------------------------+--------------+
|    0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_00 |
|    1 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_01 |
|    2 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_02 |
|    3 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_03 |
|    4 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_04 |
|    5 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_05 |
|    6 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_06 |
|    7 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_07 |
|    8 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_08 |
|    9 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_09 |
|   10 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_10 |
|   11 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_11 |
|   12 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_00 |
|   13 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_01 |
|   14 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_02 |
|   15 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_03 |
|   16 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_04 |
|   17 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_05 |
|   18 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_06 |
|   19 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_07 |
|   20 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_08 |
|   21 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_09 |
|   22 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_10 |
|   23 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_11 |
...
|   84 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_00 |
|   85 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_01 |
|   86 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_02 |
|   87 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_03 |
|   88 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_04 |
|   89 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_05 |
|   90 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_06 |
|   91 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_07 |
|   92 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_08 |
|   93 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_09 |
|   94 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_10 |
|   95 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_11 |
+------+-------------------------------------------------------------------+--------------+
96 rows in set (0.02 sec)
```

According to the sharding rule of the logical table, you can see that the database sharding process is calculating the hash function by using userId as the shard key, and the table sharding process is calculating the MM function by using actionDate as the shard key.

```
mysql> show rule from user_log2;
+------+------------+-----------+-----------------+--------------------+--------------------+---
--------------+--------------------+--------------------+
| ID   | TABLE_NAME | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT | TB
_PARTITION_KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+------+------------+-----------+-----------------+--------------------+--------------------+---
--------------+--------------------+--------------------+
|    0 | user_log2  |         0 | userId           | hash               | 8                  | ac
tionDate       | mm                 | 12                 |
+------+------------+-----------+-----------------+--------------------+--------------------+---
--------------+--------------------+--------------------+
1 row in set (0.00 sec)
```

- Create a table that is split into table shards in database shards. The database sharding process is calculating the hash function by using userId as the shard key, and the table sharding process is calculating DAY_OF_MONTH through the DD(actionDate) function and then splitting the table into table shards, with one month counted as 31 days.

  For example, if the value in the actionDate column is 2017-02-27, the value obtained by calculating the DD(actionDate) function is 27. In this case, the record is stored in table shard 27, because 27 % 31 = 27. This table shard is located in a database shard and is named user_log_27.

  ```
  CREATE TABLE user_log3(
  userId int,
  name varchar(30),
  operation varchar(30),
  actionDate DATE
  ) dbpartition by hash(userId) tbpartition by DD(actionDate) tbpartitions 31;
  ```

  According to the node topology of the logical table, you can see that 31 table shards are created in each database shard, because one month is counted as 31 days in the function. The responses are very long, and therefore are omitted by using an ellipsis (...).

```
mysql> show topology from user_log3;
+------+-------------------------------------------------------------------+--------------+
| ID   | GROUP_NAME                                                        | TABLE_NAME   |
+------+-------------------------------------------------------------------+--------------+
|    0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_00 |
|    1 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_01 |
|    2 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_02 |
|    3 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_03 |
|    4 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_04 |
|    5 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_05 |
|    6 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_06 |
|    7 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_07 |
|    8 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_08 |
|    9 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_09 |
|   10 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_10 |
|   11 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_11 |
|   12 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_12 |
|   13 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_13 |
|   14 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_14 |
|   15 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_15 |
|   16 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_16 |
|   17 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_17 |
|   18 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_18 |
|   19 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_19 |
|   20 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_20 |
|   21 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_21 |
|   22 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_22 |
|   23 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_23 |
|   24 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_24 |
|   25 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_25 |
|   26 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_26 |
|   27 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_27 |
|   28 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_28 |
|   29 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_29 |
|   30 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_30 |
...
|  237 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_20 |
|  238 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_21 |
|  239 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_22 |
|  240 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_23 |
|  241 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_24 |
|  242 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_25 |
|  243 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_26 |
|  244 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_27 |
|  245 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_28 |
|  246 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_29 |
|  247 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_30 |
+------+-------------------------------------------------------------------+--------------+
248 rows in set (0.01 sec)
```

According to the sharding rule of the logical table, you can see that the database sharding process is calculating the hash function by using userId as the shard key, and the table sharding process is calculating the DD function by using actionDate as the shard key.

```
mysql> show rule from user_log3;
+------+------------+-----------+------------------+---------------------+--------------------+---
--------------+---------------------+--------------------+
| ID   | TABLE_NAME | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT | TB
_PARTITION_KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+------+------------+-----------+------------------+---------------------+--------------------+---
--------------+---------------------+--------------------+
|    0 | user_log3  |         0 | userId           | hash                | 8                  | ac
tionDate       | dd                  | 31                 |
+------+------------+-----------+------------------+---------------------+--------------------+---
--------------+---------------------+--------------------+
1 row in set (0.01 sec)
```

- Create a table that is split into table shards in database shards. The database sharding process is calculating the hash function by using userId as the shard key, and the table sharding process is calculating DAY_OF_YEAR % 365 through the MMDD(actionDate) tbpartitions 365 function and then splitting the table into 365 physical tables, with one year counted as 365 days.

  For example, if the value in the actionDate column is 2017-02-27, the value obtained by calculating the MMDD(actionDate) function is 58. In this case, the record is stored in table shard 58. This table shard is located in a database shard and is named user_log_58.

  ```
  CREATE TABLE user_log4(
  userId int,
  name varchar(30),
  operation varchar(30),
  actionDate DATE
  ) dbpartition by hash(userId) tbpartition by MMDD(actionDate) tbpartitions 365;
  ```

  According to the node topology of the logical table, you can see that 365 table shards are created in each database shard, because one year is counted as 365 days in the function. The responses are very long, and therefore are omitted by using an ellipsis (...).

```
mysql> show topology from user_log4;
+------+-------------------------------------------------------------------+---------------+
| ID   | GROUP_NAME                                                        | TABLE_NAME    |
+------+-------------------------------------------------------------------+---------------+
...
| 2896 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_341 |
| 2897 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_342 |
| 2898 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_343 |
| 2899 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_344 |
| 2900 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_345 |
| 2901 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_346 |
| 2902 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_347 |
| 2903 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_348 |
| 2904 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_349 |
| 2905 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_350 |
| 2906 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_351 |
| 2907 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_352 |
| 2908 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_353 |
| 2909 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_354 |
| 2910 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_355 |
| 2911 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_356 |
| 2912 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_357 |
| 2913 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_358 |
| 2914 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_359 |
| 2915 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_360 |
| 2916 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_361 |
| 2917 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_362 |
| 2918 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_363 |
| 2919 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_364 |
+------+-------------------------------------------------------------------+---------------+
2920 rows in set (0.07 sec)
```

According to the sharding rule of the logical table, you can see that the database sharding process is calculating the hash function by using userId as the shard key, and the table sharding process is calculating the MMDD function by using actionDate as the shard key.

```
mysql> show rule from user_log4;
+------+------------+-----------+-----------------+---------------------+---------------------+---
---------------+---------------------+--------------------+
| ID   | TABLE_NAME | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT | TB
_PARTITION_KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+------+------------+-----------+-----------------+---------------------+---------------------+---
---------------+---------------------+--------------------+
|    0 | user_log4  |         0 | userId          | hash                | 8                   | ac
tionDate       | mmdd                | 365                |
+------+------------+-----------+-----------------+---------------------+---------------------+---
---------------+---------------------+--------------------+
1 row in set (0.02 sec)
```

- Create a table that is split into table shards in database shards. The database sharding process is calculating the hash function by using userId as the shard key, and the table sharding process is calculating DAY_OF_YEAR % 10 through the MMDD(actionDate) tbpartitions 10 function and then splitting the table into 10 physical tables, with one year counted as 365 days.

```
CREATE TABLE user_log5(
userId int,
name varchar(30),
operation varchar(30),
actionDate DATE
) dbpartition by hash(userId) tbpartition by MMDD(actionDate) tbpartitions 10;
```

According to the node topology of the logical table, you can see that 10 table shards are created in each database shard, because one year is counted as 365 days in the function and the table data is routed to 10 physical tables. The responses are very long, and therefore are omitted by using an ellipsis (...).

```
mysql> show topology from user_log5;
+------+-------------------------------------------------------------------+--------------+
| ID   | GROUP_NAME                                                        | TABLE_NAME   |
+------+-------------------------------------------------------------------+--------------+
|    0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_00 |
|    1 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_01 |
|    2 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_02 |
|    3 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_03 |
|    4 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_04 |
|    5 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_05 |
|    6 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_06 |
|    7 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_07 |
|    8 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_08 |
|    9 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_09 |
...
|   70 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_00 |
|   71 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_01 |
|   72 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_02 |
|   73 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_03 |
|   74 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_04 |
|   75 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_05 |
|   76 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_06 |
|   77 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_07 |
|   78 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_08 |
|   79 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_09 |
+------+-------------------------------------------------------------------+--------------+
80 rows in set (0.02 sec)
```

According to the sharding rule of the logical table, you can see that the database sharding process is calculating the hash function by using userId as the shard key, and the table sharding process is calculating the MMDD function by using actionDate as the shard key, and then routing the table data to 10 physical tables.

```
mysql> show rule from user_log5;
+------+------------+-----------+------------------+---------------------+---------------------+---
---------------+---------------------+--------------------+
| ID   | TABLE_NAME | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT | TB
_PARTITION_KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+------+------------+-----------+------------------+---------------------+---------------------+---
---------------+---------------------+--------------------+
|    0 | user_log5  |         0 | userId           | hash                | 8                   | ac
tionDate       | mmdd                | 10                 |
+------+------------+-----------+------------------+---------------------+---------------------+---
---------------+---------------------+--------------------+
1 row in set (0.01 sec)
```

13.1.11.2.5. Use the primary key as the shard key

When no shard key is specified for the sharding algorithm, the system uses the primary key as the shard key by default. The following illustrates how to use the primary key as the database shard key and the table shard key.

Use the primary key as the database shard key

```
CREATE TABLE prmkey_tbl(
id int,
name varchar(30),
primary key(id)
) dbpartition by hash();
```

Use the primary key as the database shard key and the table shard key

```
CREATE TABLE prmkey_multi_tbl(
id int,
name varchar(30),
primary key(id)
) dbpartition by hash() tbpartition by hash() tbpartitions 3;
```

13.1.11.2.6. Create a broadcast table

The BROADCAST clause is used to specify a broadcast table to be created. A broadcast table is replicated to each database shard and data consistency is ensured between the database shards by using a synchronization mechanism with a delay of several seconds. This feature allows you to route a JOIN operation from a Cloud Native Distributed Database PolarDB-X (PolarDB-X) instance to an underlying ApsaraDB RDS for MySQL instance to prevent the JOIN operation from being performed in multiple databases. Overview describes how to optimize SQL statements by using broadcast tables.

The following is an example statement for creating a broadcast table:

```
CREATE TABLE brd_tbl(
  id int,
  name varchar(30),
  primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 BROADCAST;
```

13.1.11.2.7. Other attributes of the MySQL CREATE TABLE statement

When creating table shards in database shards, you can also specify other attributes of the table shards in the MySQL CREATE TABLE statement. For example, you can specify other attributes as follows:

```
CREATE TABLE multi_db_multi_tbl(
  id int,
  name varchar(30),
  primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by hash(id) tbpartition by hash(id) tbpartitions 3;
```

13.1.11.3. ALTER TABLE statement

The syntax of the ALTER TABLE statement used to modify a table is as follows:

```
ALTER [ONLINE|OFFLINE] [IGNORE] TABLE tbl_name
    [alter_specification [, alter_specification] ...]
    [partition_options]
```

In a Distributed Relational Database Service (DRDS) instance, you can use this data definition language (DDL) statement to perform routine DDL operations, such as adding a column, adding an index, and modifying a data definition. For more information about the syntax, see MySQL CREATE TABLE Statement.

> **Note** If you need to modify a table shard, you are not allowed to modify the shard key.

- Add a column:

```
ALTER TABLE user_log
  ADD COLUMN idcard varchar(30);
```

- Add an index:

```
ALTER TABLE user_log
  ADD INDEX idcard_idx (idcard);
```

- Delete an index:

```
ALTER TABLE user_log
  DROP INDEX idcard_idx;
```

- Modify a field:

```
ALTER TABLE user_log
  MODIFY COLUMN idcard varchar(40);
```

13.1.11.4. DROP TABLE statement

The syntax of the DROP TABLE statement used to delete a table is as follows:

```
DROP [TEMPORARY] TABLE [IF EXISTS]
    tbl_name [, tbl_name] ...
    [RESTRICT | CASCADE]
```

The DROP TABLE statement in a Distributed Relational Database Service (DRDS) instance is the same as the DROP TABLE statement in a MySQL database. After the statement is executed, the system automatically deletes the corresponding physical table. For more information about the syntax, see MySQL DROP TABLE Statement.

For example, you can run the following statement to delete the user_log table:

```
DROP TABLE user_log;
```

13.1.11.5. FAQ about DDL statements

What can I do if an error occurs during table creation?

Data definition language (DDL) statements in a instance are processed in a distributed manner. If an error occurs, the structures of all table shards are inconsistent from each other. Therefore, you need to perform manual cleanup.

Perform the following steps:

1. Check the basic error descriptions provided by the instance, such as syntax errors. If the error message is too

long, the system will prompt you to call the SHOW WARNINGS command to view the failure cause of each database shard.

2. Run the SHOW TOPOLOGY command to view the topology of physical tables.

```
SHOW TOPOLOGY FROM multi_db_multi_tbl;
+------+-----------------+-----------------------+
| ID   | GROUP_NAME      | TABLE_NAME            |
+------+-----------------+-----------------------+
|    0 | corona_qatest_0 | multi_db_multi_tbl_00 |
|    1 | corona_qatest_0 | multi_db_multi_tbl_01 |
|    2 | corona_qatest_0 | multi_db_multi_tbl_02 |
|    3 | corona_qatest_1 | multi_db_multi_tbl_03 |
|    4 | corona_qatest_1 | multi_db_multi_tbl_04 |
|    5 | corona_qatest_1 | multi_db_multi_tbl_05 |
|    6 | corona_qatest_2 | multi_db_multi_tbl_06 |
|    7 | corona_qatest_2 | multi_db_multi_tbl_07 |
|    8 | corona_qatest_2 | multi_db_multi_tbl_08 |
|    9 | corona_qatest_3 | multi_db_multi_tbl_09 |
|   10 | corona_qatest_3 | multi_db_multi_tbl_10 |
|   11 | corona_qatest_3 | multi_db_multi_tbl_11 |
+------+-----------------+-----------------------+
12 rows in set (0.21 sec)
```

3. Run the `CHECK TABLE tablename` command to check whether the logical table has been created. For example, the following response indicates that a physical table corresponding to the logical table multi_db_multi_tbl failed to be created.

```
mysql> check table multi_db_multi_tbl;
+-------------------------------------------------+-------+----------+---------------------------
-----------------------------------------------+
| TABLE                                           | OP    | MSG_TYPE | MSG_TEXT
|
+-------------------------------------------------+-------+----------+---------------------------
-----------------------------------------------+
| andor_mysql_qatest. multi_db_multi_tbl | check | Error    | Table 'corona_qatest_0. multi_db_mu
lti_tbl_02' doesn't exist |
+-------------------------------------------------+-------+----------+---------------------------
-----------------------------------------------+
1 row in set (0.16 sec)
```

4. Continue to create or delete the table in idempotent mode to create or delete the remaining physical tables.

```
CREATE TABLE IF NOT EXISTS table1
(id int, name varchar(30), primary key(id))
dbpartition by hash(id);
DROP TABLE IF EXISTS table1;
```

What can I do if I failed to create an index or add a column?

The method for handling the failure in creating an index or adding a column is similar to that for the failure in creating a table. For more information, see Handle DDL exceptions.

13.1.11.6. DDL functions for sharding

13.1.11.6.1. Overview

is a database service that supports both database sharding and table sharding.

Support for database sharding and table sharding

The following table lists the support for database sharding and table sharding in data definition language (DDL) sharding functions.

Sharding function	Description	Support for database sharding	Support for table sharding
HASH	Performs a simple modulus operation.	Yes	Yes
UNI_HASH	Performs a simple modulus operation.	Yes	Yes
RIGHT_SHIFT	Shifts the value to the right.	Yes	Yes
RANGE_HASH	Performs double hashing.	Yes	Yes
MM	Performs hashing by month.	No	Yes
DD	Performs hashing by date.	No	Yes
WEEK	Performs hashing by week.	No	Yes
MMDD	Performs hashing by month and date.	No	Yes
YYYYMM	Performs hashing by year and month.	Yes	Yes
YYYYWEEK	Performs hashing by year and week.	Yes	Yes
YYYYDD	Performs hashing by year and date.	Yes	Yes
YYYYMM_OPT	Performs optimized hashing by year and month.	Yes	Yes
YYYYWEEK_OPT	Performs optimized hashing by year and week.	Yes	Yes
YYYYDD_OPT	Performs optimized hashing by year and date.	Yes	Yes

Note When using database sharding and table sharding in , note the following:

- In a instance, the sharding method of a logical table is defined jointly by a sharding function and a shard key. The sharding function contains the number of shards to be created and the routing algorithm. The shard key also specifies the MySQL data type of the shard key.
- When the database sharding function is the same as the table sharding function and the database shard key is the same as the table shard key in a instance, the same sharding method is used for database sharding and table sharding. This allows the instance to uniquely locate one physical table in a physical database based on the value of the shard key.
- If the database sharding method and the table sharding method of a logical table are different and an SQL query does not contain both database shard key and table shard key, the instance scans all database shards or all table shards when processing the SQL query.

Support for data types of PolarDB-X DDL sharding functions

Different DDL sharding functions support different data types. The following table lists the support for various data types in sharding functions (√ indicates supported and × indicates not supported).

Support for data types in DDL sharding functions

Sharding function	BIGINT	INT	MEDIUMINT	SMALLINT	TINYINT	VARCHAR	CHAR	DATE	DATETIME	TIMESTAMP	Other types
HASH	√	√	√	√	√	√	√	×	×	×	×
UNI_HASH	√	√	√	√	√	√	√	×	×	×	×
RANGE_HASH	√	√	√	√	√	√	√	×	×	×	×
RIGHT_SHIFT	√	√	√	√	√	×	×	×	×	×	×
MM	×	×	×	×	×	×	×	√	√	√	×
DD	×	×	×	×	×	×	×	√	√	√	×
WEEK	×	×	×	×	×	×	×	√	√	√	×
MMDD	×	×	×	×	×	×	×	√	√	√	×
YYYYMM	×	×	×	×	×	×	×	√	√	√	×
YYYYWEEK	×	×	×	×	×	×	×	√	√	√	×
YYYYDD	×	×	×	×	×	×	×	√	√	√	×
YYYYMM_OPT	×	×	×	×	×	×	×	√	√	√	×
YYYYWEEK_OPT	×	×	×	×	×	×	×	√	√	√	×
YYYYDD_OPT	×	×	×	×	×	×	×	√	√	√	×

Syntax description for PolarDB-X DDL sharding functions

is compatible with the CREATE TABLE statement in MySQL, and additionally provides the `drds_partition_options` keyword to support database sharding and table sharding:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    (create_definition,...)
    [table_options]
    [drds_partition_options]
    [partition_options]
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    [(create_definition,...)]
    [table_options]
    [drds_partition_options]
    [partition_options]
    select_statement
drds_partition_options:
    DBPARTITION BY
        { {HASH|YYYYMM|YYYYWEEK|YYYYDD|YYYYMM_OPT|YYYYWEEK_OPT|YYYYDD_OPT}([column])}
        [TBPARTITION BY
            { {HASH|MM|DD|WEEK|MMDD|YYYYMM|YYYYWEEK|YYYYDD|YYYYMM_OPT|YYYYWEEK_OPT|YYYYDD_OPT}(column
)}
          [TBPARTITIONS num]
        ]
```

13.1.11.6.2. HASH

Requirements

- The shard key must be an integer or a string.
- This sharding function has no requirements on the version of a Distributed Relational Database Service (DRDS)

instance. It supports all DRDS instances by default.

Routing method

When the HASH function is run by using different shard keys for database sharding and table sharding, perform the remainder operation on the value of the database shard key based on the number of database shards. If the value of the shard key is a string, the string is converted to a hash value before route calculation. For example, HASH('8') is equivalent to 8 % D, where D indicates the number of database shards.

When the UNI_HASH function is run by using the same shard key for both database sharding and table sharding, perform the remainder operation on the value of the shard key based on the total number of table shards. For example, assume that two database shards are created, each database shard contains four table shards, table shards 0 to 3 are stored in database shard 0, and table shards 4 to 7 are stored in database shard 1. If a key value is 15, the key value 15 is distributed to table shard 7 in database shard 1, because 15 % (2 x 4) = 7.

Scenarios

- HASH is applicable when database sharding is implemented by user ID or order ID.
- HASH is also applicable when the shard key is a string.

Examples

If you need to create a non-partition table in database shards by using the HASH function based on the ID column, you can use the following CREATE TABLE statement:

```
create table test_hash_tb (
    id int,
    name varchar(30) DEFAULT NULL,
    create_time datetime DEFAULT NULL,
    primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by HASH(ID);
```

Notes

The HASH is a simple modulus operation. The output distribution of the HASH function can be even only when the values in the shard column are evenly distributed.

13.1.11.6.3. UNI_HASH

Requirements

- The shard key must be an integer or a string.
- The version of the instance must be 5.1.28-1508068 or later. For more information about the release notes, see View the instance version.

Routing method

When the UNI_HASH function is used for database sharding, perform a remainder operation on the value of the database shard key based on the number of database shards. If the value of the shard key is a string, the string is converted to a hash value before route calculation. For example, HASH('8') is equivalent to 8 % D, where D indicates the number of database shards.

When the UNI_HASH function is run by using the same shard key for both database sharding and table sharding, perform the remainder operation on the value of the database shard key based on the number of database shards first (this step is different from that in the HASH function). Then, the data is evenly distributed to the table shards in the database shard.

Scenarios

- UNI_HASH is applicable when database sharding is implemented by user ID or order ID.

- UNI_HASH is also applicable when the shard key is an integer or a string.
- UNI_HASH can be used when the following conditions are met: Two logical tables need to be partitioned into different numbers of table shards in database shards based on the same shard key. In addition, the two tables are frequently joined by using a JOIN statement based on the shard key.

Comparison with HASH

When you use the UNI_HASH function to create a non-partition table in database shards, the routing method is the same as that used in the HASH function. Specifically, the route is calculated by performing the remainder operation on the key value of the database shard key based on the number of database shards.

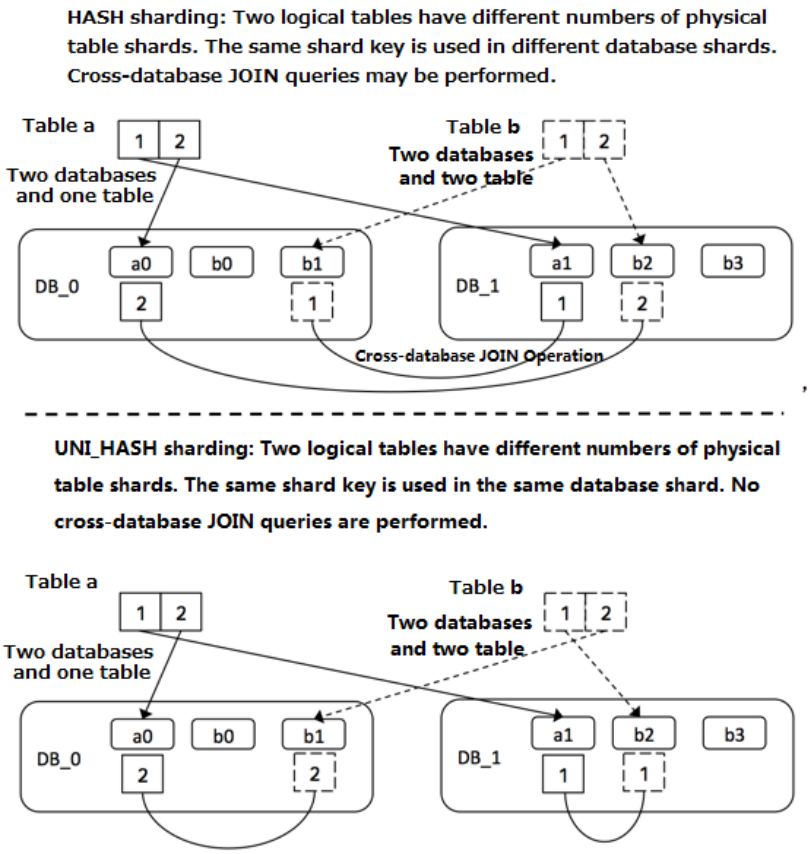
When the UNI_HASH function is run by using the same shard key for both database sharding and table sharding, as the number of table shards changes, the database shard route calculated based on the same key value may also change.

When the UNI_HASH function is run by using the same shard key for both database sharding and table sharding, the database shard route calculated based on the same key value is always the same regardless of the number of table shards.

If two logical tables need to be partitioned into different table shards in database shards based on the same shard key, when the two tables are joined by using the HASH function based on the shard key, multi-database join may occur. However, when the two tables are joined by using the UNI_HASH function based on the shard key, multi-database join does not occur.

Assume that you have two database shards and two logical tables, and each database shard in logical table a stores one table shard and each database shard in logical table b stores two table shards. The following figures separately show the results of a JOIN query for logical tables a and b after the HASH function is used for sharding and the results of a JOIN query for logical tables a and b after the HASH function is used for sharding.

Comparison between HASH and UNI_HASH



Examples

If you need to create four table shards in each database shard by using the UNI_HASH function based on the ID column, you can run the following CREATE TABLE statement:

```
create table test_hash_tb (
    id int,
    name varchar(30) DEFAULT NULL,
    create_time datetime DEFAULT NULL,
    primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
dbpartition by UNI_HASH(ID)
tbpartition by UNI_HASH(ID) tbpartitions 4;
```

Precautions

The UNI_HASH is a simple modulus operation. The output distribution of the UNI_HASH function can be even only when the values in the shard column are evenly distributed.

13.1.11.6.4. RIGHT_SHIFT

Requirements

- The shard key must be an integer.
- The version of the instance must be 5.1.28-1320920 or later. For more information, see View the instance version.

Routing method

Shift the value of the database shard key to the right by a specified number of binary digits, and then perform the remainder operation on the obtained integer based on the number of database shards or table shards. In particular, you can specify the number of shifted digits by running a data definition language (DDL) statement.

Scenarios

RIGHT_SHIFT is applicable to improve the evenness of the hash results when the lower-digit parts of most shard key values are very similar to each other but the higher-digit parts vary greatly.

Assume that four shard key values are available: 12340000, 12350000, 12460000, and 12330000. The four lower digits of the four values are all 0000. Directly hashing the values of the shard keys outputs poor results. However, if you run the RIGHT_SHIFT(shardKey, 4) statement to shift the values of the shard keys to the right by four digits, to obtain 1234, 1235, 1246, and 1233, the hashing results are improved.

Examples

If you need to use the ID column as a shard key and shift the values of the ID column to the right by four binary digits to obtain hash values, you can run the following CREATE TABLE statement:

```
create table test_hash_tb (
    id int,
    name varchar(30) DEFAULT NULL,
    create_time datetime DEFAULT NULL,
    primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
dbpartition by RIGHT_SHIFT(id, 4)
tbpartition by RIGHT_SHIFT(id, 4) tbpartitions 2;
```

Precautions

The number of shifted digits cannot exceed the number of digits occupied by the integer.

13.1.11.6.5. RANGE_HASH

Requirements

- The shard key must be a character or a number.
- The version of the instance must be 5.1.28-1320920 or later. For more information, see View the instance version.

Routing method

Calculate the hash value based on the last N digits of any shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes the route computing. The number N is the third parameter in the function.

For example, during calculation of the RANGE_HASH(COL1, COL2, N) function, COL1 is preferentially selected and then truncated to obtain the last N digits for calculation. If COL1 does not exist, COL2 is selected and truncated for calculation.

Scenarios

RANGE_HASH is applicable to scenarios where a table needs to be partitioned by two shard keys but query is performed only based on the value of one shard key.

Examples

Assume that a database is partitioned into eight physical databases. Our customer has the following requirements:

The order table of a business needs to be partitioned into database shards by buyer ID and order ID. The query is executed based on either the buyer ID or order ID as the condition.

In this case, you can run the following DDL statement to create the order table:

```
create table test_order_tb (
    id int,
    buyer_id varchar(30) DEFAULT NULL,
    order_id varchar(30) DEFAULT NULL,
    create_time datetime DEFAULT NULL,
    primary key(id)
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8
    dbpartition by RANGE_HASH(buyer_id,order_id, 10)
    tbpartition by RANGE_HASH (buyer_id,order_id, 10) tbpartitions 3;
```

Precautions

- Neither of the two shard keys can be modified.
- Data insertion fails if the two shard keys point to different database shards or table shards.

13.1.11.6.6. MM

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- MM is only applicable to table sharding.
- The version of the instance must be 5.1.28-1320920 or later. For more information, see View the instance version.

Routing method

Perform the remainder operation based on the month that corresponds to the time value of the database shard key to obtain the table shard subscript.

Scenarios

MM can be used to partition tables by month. The table shard name indicates a specific month.

Examples

Assume that we need to perform database sharding by ID, perform table sharding for the create_time column by month, and map every month to a physical table. The data definition language (DDL) statement is as follows:

```
create table test_mm_tb (
    id int,
    name varchar(30) DEFAULT NULL,
    create_time datetime DEFAULT NULL,
    primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
dbpartition by HASH(id)
tbpartition by MM(create_time) tbpartitions 12;
```

Precautions

When you partition tables with MM, ensure that each database shard has no more than 12 table shards because a year has 12 months.

13.1.11.6.7. DD

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- DD is only applicable to table sharding.
- The version of the instance must be 5.1.28-1320920 or later. For more information, see View the instance version.

Routing method

Perform the remainder operation based on the day of the month that corresponds to the time value of the database shard key to obtain the table shard subscript.

Scenarios

DD can be used to partition tables based on a specified number of days in a month, that is, a date. The subscript of the table shard name indicates the day in a month. A month has 31 days at most.

Examples

Assume that we need to perform database sharding by ID, perform table sharding for the create_time column by day, and map every day to a physical table. The data definition language (DDL) statement is as follows:

```
create table test_dd_tb (
    id int,
    name varchar(30) DEFAULT NULL,
    create_time datetime DEFAULT NULL,
    primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
dbpartition by HASH(id)
tbpartition by DD(create_time) tbpartitions 31;
```

Precautions

When you partition tables with DD, ensure that each database shard has no more than 31 table shards because a month has 31 days at most.

13.1.11.6.8. WEEK

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- WEEK is only applicable to table sharding.
- The version of the instance must be 5.1.28-1320920 or later. For more information, see View the instance version.

Routing method

Perform the remainder operation based on the day of a week that corresponds to the time value of the database shard key to obtain the table shard subscript.

Scenarios

WEEK can be used to partition tables based on days in a week. The subscript of the table shard name corresponds to each day of a week, from Monday to Sunday.

Examples

Assume that we need to perform database sharding by ID, perform table sharding for the create_time column by week, and map every day of a week (from Monday to Sunday) to a physical table. The data definition language (DDL) statement is as follows:

```
create table test_week_tb (
    id int,
    name varchar(30) DEFAULT NULL,
    create_time datetime DEFAULT NULL,
    primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
dbpartition by HASH(name)
tbpartition by WEEK(create_time) tbpartitions 7;
```

Precautions

When you partition tables with WEEK, ensure that each database shard has no more than seven table shards because a week has seven days.

13.1.11.6.9. MMDD

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- MMDD is only applicable to table sharding.
- The version of the instance must be 5.1.28-1320920 or later. For more information, see View the instance version.

Routing method

Perform the remainder operation based on the number of days in a year that corresponds to the time value of the database shard key to obtain the table sharding subscript.

Scenarios

MMDD can be used to partition tables based on the number of days in a year that corresponds to a date in that year. The subscript of the table shard name indicates the day in that year, with a maximum of 366 days in a year.

Examples

Assume that we need to perform database sharding by ID, create tables for the create_time column by date (month-day), and map every day of a year to a physical table. The data definition language (DDL) statement is as follows.

```
create table test_mmdd_tb (
    id int,
    name varchar(30) DEFAULT NULL,
    create_time datetime DEFAULT NULL,
    primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
dbpartition by HASH(name)
tbpartition by MMDD(create_time) tbpartitions 365;
```

Precautions

When you partition tables with MMDD, ensure that each database shard has no more than 366 table shards because a year has 366 days at most.

13.1.11.6.10. YYYYMM

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- The version of the instance must be 5.1.28-1320920 or later. For more information, see View the instance version.

Routing method

Calculate the hash value based on the year and months of the year in the time value of the database shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes route computing.

For example, YYYYMM('2012-12-31 12:12:12') is equivalent to (2012 x 12 + 12) % D, where D indicates the number of database shards.

Scenarios

YYYYMM can be used to partition databases by year and month. We recommend that you use YYYYMM with tbpartition YYYYMM(ShardKey).

Assume that a database is partitioned into eight physical databases. Our customer has the following requirements:

- Perform database sharding for a service by year and month.
- Distribute data from every month within two years to a separate table shard.
- Distribute a query with the database and table shard keys to a physical table shard of a physical database shard.

The preceding requirements can be met by using YYYYMM. For the requirement of distributing data from every month within two years to a table shard (that is, one table shard stores the data of one month), create at least 24 physical table shards because a year has 12 months. Create three physical table shards for each database shard because the instance contains eight database shards. The data definition language (DDL) statement is as follows.

```
create table test_yyyymm_tb (
        id int,
        name varchar(30) DEFAULT NULL,
        create_time datetime DEFAULT NULL,
        primary key(id)
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8
    dbpartition by YYYYMM(create_time)
    tbpartition by YYYYMM(create_time) tbpartitions 3;
```

Precautions

- YYYYMM does not support distributing data from every month in every year to a separate table shard. Instead, the number of table shards must be fixed.
- After a cycle over months (for example, a cycle exists between 2012-03 and 2013-03), data from the same month may be routed to the same database or table shard, depending on the actual number of table shards.

13.1.11.6.11. YYYYWEEK

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- The version of the instance must be 5.1.28-1320920 or later. For more information, see View the instance version.

Routing method

Calculate the hash value based on the year and weeks of the year in the time value of the database shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes route computing.

For example, YYYYWEEK('2012-12-31 12:12:12') is equivalent to (2013 x 52 + 1) % D, with the date 2012-12-31 falling on the first week of 2013, where D indicates the number of database shards.

Scenarios

YYYYWEEK can be used to partition databases by year and the number of weeks in a year. We recommend that you use YYYYWEEK with tbpartition YYYYWEEK(ShardKey).

Assume that a database is partitioned into eight physical databases. Our customer has the following requirements:

- Perform database sharding for a service by year and by week.
- Distribute data from every week within two years to a separate table shard.
- Distribute a query with the database and table shard keys to a physical table shard of a physical database shard.

The preceding requirements can be met by using YYYYWEEK. For the requirement of distributing data from every week within two years to a table shard (that is, one table shard stores the data of one week), create at least 106 physical table shards because a year has roughly 53 weeks (rounded). Create 14 physical table shards for each database shard because the instance contains eight database shards (14 x 8 = 112 > 106). We recommend that the number of table shards be an integer multiple of the number of database shards. The data definition language (DDL) statement is as follows:

```
create table test_yyyymm_tb (
        id int,
        name varchar(30) DEFAULT NULL,
        create_time datetime DEFAULT NULL,
        primary key(id)
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8
    dbpartition by YYYYWEEK(create_time)
    tbpartition by YYYYWEEK(create_time) tbpartitions 14;
```

Precautions

- YYYYWEEK does not support distributing data from every week in every year to a separate table shard. Instead, the number of table shards must be fixed.
- After a cycle over weeks (for example, a cycle exists between the first week of 2012 and the first week of 2013), data from the same week after a cycle may be routed to the same database shard or table shard, depending on the actual number of table shards.

13.1.11.6.12. YYYYDD

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- The version of the instance must be 5.1.28-1320920 or later. For more information, see View the instance version.

Routing method

Calculate the hash value based on the year and days of the year in the time value of the database shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes route computing.

For example, YYYYDD('2012-12-31 12:12:12') is equivalent to (2012 x 366 + 365) % D, with 2012-12-31 as the 365th day of 2012, where D indicates the number of database shards.

Scenarios

Database sharding is performed by year and the number of days in a year. We recommend that you use YYYYDD with tbpartition YYYYDD(ShardKey).

Assume that a database is partitioned into eight physical databases. Our customer has the following requirements:

- Perform database sharding for a service by year and day.
- Distribute data from every week within two years to a separate table shard.
- Distribute a query with the database and table shard keys to a physical table shard of a physical database shard.

The preceding requirements can be met by using YYYYDD. For the requirement of distributing data from every day within two years to a table shard (that is, one table shard stores the data of one day), create at least 732 physical table shards because a year has up to 366 days. Create 92 physical table shards for each database shard because the instance contains eight database shards (732/8 = 91.5, rounded to 92). We recommend that the number of table shards be an integer multiple of the number of database shards. The data definition language (DDL) statement is as follows:

```
create table test_yyyydd_tb (
        id int,
        name varchar(30) DEFAULT NULL,
        create_time datetime DEFAULT NULL,
        primary key(id)
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8
    dbpartition by YYYYDD(create_time)
    tbpartition by YYYYDD(create_time) tbpartitions 92;
```

Precautions

- YYYYDD does not support distributing data from every day in every year to a separate table shard. Instead, the number of table shards must be fixed.
- After a cycle of a specific date (for example, a cycle exists between 2012-03-01 and 2013-03-01), data from the same date may be routed to the same database shard or table shard, depending on the actual number of table shards.

13.1.11.6.13. YYYYMM_OPT

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- The year and month of user data increase naturally over time, rather than randomly.
- The version of the instance must be 5.1.28-1320920 or later. For more information, see View the instance version.

Optimizations

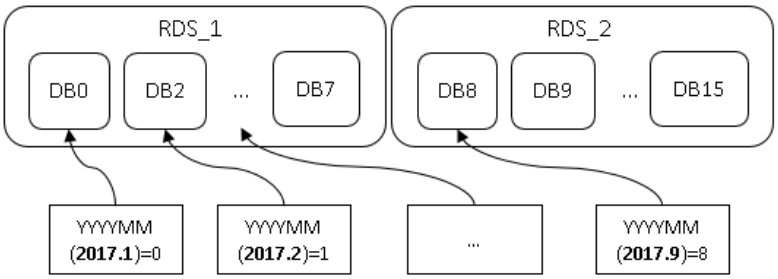
Compared with YYYYMM, YYYYMM_OPT maintains the even distribution of data among ApsaraDB RDS for MySQL instances as the timeline increases.

For example, assume that two ApsaraDB RDS for MySQL instances are attached to a instance, with 16 database shards. DB0 to DB7 shards are located on one ApsaraDB RDS for MySQL instance, and DB8 to DB15 shards are located on the other ApsaraDB RDS for MySQL instance.

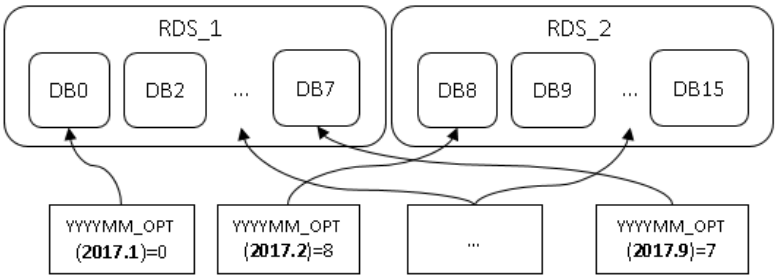
The following figure shows the mappings when YYYYMM and YYYYMM_OPT are used for database sharding, respectively.

Comparison between YYYYMM and YYYYMM_OPT

As the time goes on linearly, YYYYMM fills data in ApsaraDB for RDS instances in sequence (data is first distributed to the database shards of RDS_1, then to the database shards of RDS_2, and then to the database shards of RDS_1 again).



YYYYMM_OPT evenly distributes data between ApsaraDB for RDS instances as the time goes on (data is alternately distributed between RDS_1 and RDS_2, so that the data size of the two RDS instances is balanced).



- YYYYMM_OPT distributes data evenly to each ApsaraDB RDS for MySQL instance, helping to maximize the performance of each ApsaraDB RDS for MySQL instance.
- How to choose between YYYYMM and YYYYMM_OPT:
  - YYYYMM_OPT can be used to distribute data evenly to each ApsaraDB RDS for MySQL instance if the time of service data generation increases sequentially and the data volume does not differ much between the time points.
  - YYYYMM is applicable if the time of data generation increases randomly rather than sequentially.

Routing method

- Calculate the hash value based on the year and months of the year in the time value of the database shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes route computing.
- The hash calculation based on the database and table shard key considers the data distribution among the ApsaraDB RDS for MySQL instances that connect to the instances.

Scenarios

- Databases and tables need to be partitioned by year and month, respectively.
- Data must be evenly distributed to each ApsaraDB RDS for MySQL instance that connects to the instance.
- The time of the shard key increases sequentially rather than randomly, and the data volume is relatively average from month to month. For example, the number of monthly journal logs increases every month, and the log data is not concentrated on the same ApsaraDB RDS for MySQL instance.

Precautions

- YYYYMM_OPT does not support distributing data from every month in every year to a separate table shard. Instead, the number of table shards must be fixed.
- After a cycle over months (for example, a cycle exists between 2012-03 and 2013-03), data from the same month may be routed to the same database or table shard, depending on the actual number of table shards.

13.1.11.6.14. YYYYWEEK_OPT

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- The version of the instance must be 5.1.28-1320920 or later. For more information, see View the instance version.

Optimizations

- Compared with YYYYWEEK, YYYYWEEK_OPT maintains the even distribution of data among ApsaraDB RDS for MySQL instances as the timeline increases. The effect is similar to YYYYMM_OPT.
- YYYYWEEK_OPT distributes data evenly to each ApsaraDB RDS for MySQL instance, helping to maximize the performance of each ApsaraDB RDS for MySQL instance.
- How to choose between YYYYWEEK and YYYYWEEK_OPT:
  - YYYYWEEK_OPT can be used to distribute data evenly to each ApsaraDB RDS for MySQL instance if the time of service data increases sequentially and the data volume does not differ much between time points.
  - YYYYWEEK is applicable if the time of data generation increases randomly rather than sequentially.

Routing method

- Calculate the hash value based on the year and weeks of the year in the time value of the database shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes route computing.
- The hash calculation based on the database and table shard key considers the data distribution among the ApsaraDB RDS for MySQL instances that connect to the instances.

Scenarios

- Databases and tables are partitioned by year and week, respectively.
- Data must be evenly distributed to each ApsaraDB RDS for MySQL instance that connects to the instance.
- The time of the shard key increases sequentially rather than randomly, and the data volume is relatively average from week to week. For example, the number of weekly journal logs increases every week, and the log data is not concentrated on the same ApsaraDB RDS for MySQL instance.

Precautions

- YYYYWEEK_OPT does not support distributing data from every week in every year to a separate table shard. Instead, the number of table shards must be fixed.
- After a cycle over weeks (for example, a cycle exists between the first week of 2012 and the first week of 2013), data from the same week after a cycle may be routed to the same database shard or table shard, depending on the actual number of table shards.

13.1.11.6.15. YYYYDD_OPT

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- The version of the instance must be 5.1.28-1320920 or later. For more information, see View the instance version.

Optimizations

- Compared with YYYYDD, YYYYDD_OPT maintains the even distribution of data among ApsaraDB RDS for MySQL instances as the timeline increases. The effect is similar to YYYYMM_OPT.
- YYYYDD_OPT distributes data evenly to each ApsaraDB RDS for MySQL instance, helping to maximize the performance of each ApsaraDB RDS for MySQL instance.
- How to choose between YYYYDD and YYYYDD_OPT:
  - YYYYDD_OPT can be used to distribute data evenly to each ApsaraDB RDS for MySQL instance if the time of service data generation increases sequentially and the data volume does not differ much between time points.
  - YYYYDD is applicable if the time of data generation increases randomly rather than sequentially.

Routing method

- Calculate the hash value based on the year and days of the year in the time value of the database shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes route computing.
- The hash calculation based on the database and table shard key considers the data distribution among the ApsaraDB RDS for MySQL instances that connect to the instances.

Scenarios

- Databases and tables need to be partitioned by year and by day, respectively.
- Data must be evenly distributed to each ApsaraDB RDS for MySQL instance that connects to the instance.
- The time of the shard key increases sequentially rather than randomly, and the data volume is relatively average from day to day. For example, the number of daily journal logs increases every day, and the log data is not concentrated on the same ApsaraDB RDS for MySQL instance.

Precautions

- YYYYDD_OPT does not support distributing data from every day in every year to a separate table shard. Instead, the number of table shards must be fixed.
- After a cycle of a specific date (for example, a cycle exists between 2012-03-01 and 2013-03-01), data from the same date may be routed to the same database shard or table shard, depending on the actual number of table shards.

13.1.12. Automatic protection of important SQL statements

In Distributed Relational Database Service (DRDS), the data manipulation language (DML) statements are the same as MySQL statements.

We recommend that you include the shard key in the **SELECT** and **UPDATE** statements of DRDS. The **INSERT** statement of DRDS must include the shard key and a non-empty key value.

By default, DRDS disables full-table deletion and updating to avoid misoperation.

The following statements are prohibited by default:

- A DELETE statement without the WHERE or LIMIT condition
- An UPDATE statement without the WHERE or LIMIT condition

If you need to perform full-table deletion or update, you can temporarily skip this limit by using the following hint:

```
HINT: /! TDDL:FORBID_EXECUTE_DML_ALL=false*/
```

Examples

- Full-table deletion is intercepted by default.

```
mysql> delete from tt;
ERR-CODE: [TDDL-4620][ERR_FORBID_EXECUTE_DML_ALL] Forbid execute DELETE ALL or UPDATE ALL sql. Mor
e: [http://middleware.alibaba-inc.com/faq/faqByFaqCode.html?faqCode=TDDL-4620]
```

  The operation is successful if the following hint is added:

```
mysql> /! TDDL:FORBID_EXECUTE_DML_ALL=false*/delete from tt;
Query OK, 10 rows affected (0.21 sec)
```

- Full-table update is intercepted by default.

```
mysql> update tt set id = 1;
ERR-CODE: [TDDL-4620][ERR_FORBID_EXECUTE_DML_ALL] Forbid execute DELETE ALL or UPDATE ALL sql. Mor
e: [http://middleware.alibaba-inc.com/faq/faqByFaqCode.html?faqCode=TDDL-4620]
```

  The operation is successful if the following HINT is added:

```
mysql> /! TDDL:FORBID_EXECUTE_DML_ALL=false*/update tt set id = 1;
Query OK, 10 rows affected (0.21 sec)
```

- This limit does not apply to DELETE or UPDATE statements that contain the WHERE or LIMIT condition.

```
mysql> delete from tt where id = 1;
Query OK, 1 row affected (0.21 sec)
```

13.1.13. PolarDB-X sequence

13.1.13.1. Overview

A Distributed Relational Database Service (DRDS) sequence is a 64-digit number that corresponds to the signed BIGINT type in MySQL. It is used to create a globally unique and sequentially incremental numeric sequence, such as the values of primary key columns and unique index columns.

DRDS sequences are used in the following two ways:

- Explicit sequences are created and maintained by using sequence-specific data definition language (DDL) syntax and can be used independently. The sequence value can be acquired by using `select seq.nextval;`, in which `seq` indicates the sequence name.
- Implicit sequences are used to automatically fill in primary keys with AUTO_INCREMENT defined and are automatically maintained by DRDS.

> **Notice** DRDS creates implicit sequences only after AUTO_INCREMENT is defined for partitioned tables and broadcast tables. This is not the case for non-partition tables. The AUTO_INCREMENT value of a non-partition table is created by ApsaraDB RDS for MySQL.

Types and features of DRDS sequences

Currently, three types of DRDS sequences are supported.

Type (abbreviation)	Globally unique	Consecutive	Monotonically increasing	Monotonically increasing within the same connection	Non-single point	Data type	Readability
Group sequence (GROUP)	Yes	No	No	Yes	Yes	All integer types	High
Time-based sequence (TIME)	Yes	No	Monotonically increasing at the macro level and non-monotonically increasing at the micro level	Yes	Yes	Only BIGINT is supported	Low
Simple sequence (SIMPLE)	Yes	Yes	Yes	Yes	No	All integer types	High

Concepts:

- **Consecutive:** If the current value is n, the next value must be n + 1. If the next value is not n + 1, it is nonconsecutive.
- **Monotonically increasing:** If the current value is n, the next value must be a number greater than n.
- **Single point:** The risk of single point of failure exists.
- **Monotonically increasing at the macro level and non-monotonically increasing at the micro level:** An example of this is `1, 3, 2, 4, 5, 7, 6, 8, ...`

Group sequence (GROUP, used by default)

Features

A group sequence is a globally unique sequence with natural numeric values, which are not necessarily consecutive or monotonically increasing. If the sequence type is not specified, DRDS uses the group sequence type by default.

- **Advantages:** A group sequence is globally unique and provides excellent performance, preventing single point of failure.
- **Disadvantages:** A group sequence may contain nonconsecutive values, which may not necessarily start from the initial value and do not cycle.

Implementation

The values of a group sequence are created by multiple nodes to ensure high availability. The values in a segment are nonconsecutive if the values are not all used, such as in the case of disconnection.

Time-based sequence

Features

A time-based sequence consists of **a timestamp, node ID, and serial number**. It is globally unique and automatically increments at the macro level. Value updates are database-independent and not persistently stored in databases. Only names and types are stored in databases. This delivers good performance to time-based sequences, which create values like 776668092129345536, 776668098018148352, 776668111578333184, and 776668114812141568.

> Notice **Sequence values must be of the BIGINT type** when used in the auto-increment columns of tables.

- **Advantages:** Time-based sequences are globally unique with good performance.
- **Disadvantages:** The values of a time-based sequence are nonconsecutive. The START WITH, INCREMENT BY, MAXVALUE, and CYCLE or NOCYCLE parameters are invalid for time-based sequences.

Simple sequence

Features

Only simple sequences support the START WITH, INCREMENT BY, MAXVALUE, and CYCLE or NOCYCLE parameters.

- **Advantages:** Simple sequences are globally unique and monotonically increasing with consecutive values.
- **Disadvantages:** Simple sequences are prone to single point of failure, poor performance, and bottlenecks. Use them with caution.

Implementation

Each sequence value must be persistently stored.

Scenarios

Group sequences, time-based sequences, and simple sequences are globally unique and can be used in **primary key columns** and **unique index columns**.

- We recommend that you use **group sequences**.
- Use only simple sequences for services that strongly depend on consecutive sequence values. Pay attention to sequence performance.
- We recommend that you use time-based sequences if you have high requirements for sequence performance, the amount of data inserted to tables is small, and large sequence values are acceptable. Time-based sequences are CPU-bound with no requirements on computing lock, database dependence, or persistent storage.

The following example shows how to create a sequence with an initial value of 100000 and a step of 1.

- A **simple sequence** creates globally unique, consecutive, and monotonically increasing values, such as 100000, 100001, 100002, 100003, 100004, ..., 200000, 200001, 200002, 200003... The values of a simple sequence are persistently stored. Even after services are restarted upon a single point of failure, values are still created consecutively from the breakpoint. However, simple sequences have poor performance because each value is persistently stored once it is created.
- A **group sequence** may create values like 200001, 200002, 200003, 200004, 100001, 100002, 100003...

> Notice
>
> - The initial value of a group sequence is not necessarily the same as the START WITH value (which is 100000 in this example) but is invariably greater than this value. In this example, the initial value is 200001.
> - A group sequence is globally unique but may contain nonconsecutive values, which may occur when a node is faulty or the connection that only uses partial values is closed. The group sequence in this example contains nonconsecutive values because the values between 200004 and 100001 are missing.

- A **time-based sequence** may create values like 776668092129345536, 776668098018148352, 776668111578333184, 776668114812141568...

13.1.13.2. Explicit sequence usage

This topic describes how to use data definition language (DDL) statements to create, modify, delete, and query sequences and how to acquire the values of explicit sequences.

Create a sequence

Syntax:

```
CREATE [ GROUP | SIMPLE | TIME ] SEQUENCE <name>
[ START WITH <numeric value> ] [ INCREMENT BY <numeric value> ]
[ MAXVALUE <numeric value> ] [ CYCLE | NOCYCLE ]
```

Parameters:

Parameter	Description	Applicable To
START WITH	The initial sequence value. If it is not set, the default value is 1.	Simple sequence and group sequence
INCREMENT BY	The increment (or interval value or step) of each sequence increase. If it is not set, the default value is 1.	Simple sequence
MAXVALUE	The maximum sequence value. If it is not specified, the default value is the maximum value of the signed BIGINT type.	Simple sequence
CYCLE or NOCYCLE	Indicates whether to repeat the sequence value which starts from the value specified by START WITH after the sequence value reaches the maximum value. If it is not specified, the default value is NOCYCLE.	Simple sequence

> **Note**
>
> - If the sequence type is not specified, the group sequence type is used by default.
> - The INCREMENT BY, MAXVALUE, and CYCLE or NOCYCLE parameters are invalid for group sequences.
> - The START WITH, INCREMENT BY, MAXVALUE, and CYCLE or NOCYCLE parameters are invalid for time-based sequences.
> - Group sequences are nonconsecutive. The START WITH parameter only provides reference for group sequences. **The initial group sequence value is not necessarily the same as but is greater than the value of START WITH.**

Example 1: Create a group sequence.

- Method 1:

```
mysql> CREATE SEQUENCE seq1;
Query OK, 1 row affected (0.27 sec)
```

- Method 2:

```
mysql> CREATE GROUP SEQUENCE seq1;
Query OK, 1 row affected (0.27 sec)
```

Example 2: Create a time-based sequence.

```
mysql> CREATE TIME SEQUENCE seq1;
Query OK, 1 row affected (0.27 sec)
```

Example 3: Create a simple sequence with an initial value of 1000, step of 2, and maximum value of 99999999999, which does not repeat after increasing to the maximum value.

```
mysql> CREATE SIMPLE SEQUENCE seq2 START WITH 1000 INCREMENT BY 2 MAXVALUE 99999999999 NOCYCLE;
Query OK, 1 row affected (0.03 sec)
```

Modify a sequence

Distributed Relational Database Service (DRDS) allows you to modify sequences in the following ways:

- For simple sequences, change the values of START WITH, INCREMENT BY, MAXVALUE, and CYCLE or NOCYCLE.
- For group sequences, change the value of START WITH.
- Convert the sequence type to another.

Syntax:

```
ALTER SEQUENCE <name> [ CHANGE TO GROUP | SIMPLE | TIME ]
START WITH <numeric value> [ INCREMENT BY <numeric value> ]
[ MAXVALUE <numeric value> ] [ CYCLE | NOCYCLE ]
```

Parameters:

Parameter	Description	Applicable To
START WITH	The initial sequence value. If it is not set, the default value is 1.	Simple sequence and group sequence
INCREMENT BY	The increment (or interval value or step) of each sequence increase. If it is not set, the default value is 1.	Simple sequence
MAXVALUE	The maximum sequence value. If it is not specified, the default value is the maximum value of the signed BIGINT type.	Simple sequence
CYCLE or NOCYCLE	Indicates whether to repeat the sequence value which starts from the value specified by START WITH after the sequence value reaches the maximum value. If it is not specified, the default value is NOCYCLE.	Simple sequence

> **Note**
>
> - Group sequences are nonconsecutive. The START WITH parameter only provides reference for group sequences. **The initial group sequence value is not necessarily the same as but is greater than the value of START WITH.**
> - If you set START WITH when modifying a simple sequence, the START WITH value takes effect immediately. The following sequence value starts from the new START WITH value. For example, if you change the START WITH value to 200 when the sequence value increases to 100, the following sequence value starts from 200.
> - Before changing the START WITH value, you need to analyze the existing sequence values and the speed of creating sequence values to avoid conflicts. Do not change the START WITH value unless necessary.

Example: Change the initial value, step, and maximum value of the simple sequence named seq2 to 3000, 5, and 1000000 respectively, and set CYCLE.

```
mysql> ALTER SEQUENCE seq2 START WITH 3000 INCREMENT BY 5 MAXVALUE 1000000 CYCLE;
Query OK, 1 row affected (0.01 sec)
```

Convert the sequence type to another

- Use the `CHANGE TO <sequence_type>` clause of `ALTER SEQUENCE`.
- If the `CHANGE TO` clause of `ALTER SEQUENCE` is specified, add the `START WITH` parameter to prevent duplicate values. This parameter is optional if `CHANGE TO` is not specified.

Example: Convert a group sequence to a simple sequence.

```
mysql> ALTER SEQUENCE seq1 CHANGE TO SIMPLE START WITH 1000000;
Query OK, 1 row affected (0.02 sec)
```

Delete a sequence

Syntax:

```
DROP SEQUENCE <name>
```

Example:

```
mysql> DROP SEQUENCE seq3;
Query OK, 1 row affected (0.02 sec)
```

Query sequences

Syntax:

```
SHOW SEQUENCES
```

Example: The TYPE column lists the sequence types in the abbreviated form.

```
mysql> SHOW SEQUENCES;
+------------+---------------------+-------------+------------+---------------------+-------+-------
-+
| NAME       | VALUE               | INCREMENT_BY | START_WITH | MAX_VALUE           | CYCLE | TYPE
|
+------------+---------------------+-------------+------------+---------------------+-------+-------
-+
| AUTO_SEQ_1 | 91820513            | 1            | 91820200   | 9223372036854775807 | N     | SIMPLE
|
| AUTO_SEQ_4 | 91820200            | 2            | 1000       | 9223372036854775807 | Y     | SIMPLE
|
| seq_test   | N/A                 | N/A          | N/A        | N/A                 | N/A   | TIME
|
| AUTO_SEQ_2 | 100000              | N/A          | N/A        | N/A                 | N/A   | GROUP
|
| AUTO_SEQ_3 | 200000              | N/A          | N/A        | N/A                 | N/A   | GROUP
|
+------------+---------------------+-------------+------------+---------------------+-------+-------
-+
5 rows in set (0.01 sec)
```

Get the sequence value

Syntax:

```
< sequence name >.NEXTVAL
```

Example:

```
SELECT sample_seq.nextVal FROM dual;
+--------------------+
| SAMPLE_SEQ.NEXTVAL |
+--------------------+
|             101001 |
+--------------------+
1 row in set (0.04 sec)
```

You can also write SAMPLE_SEQ.nextVal as a value to the SQL statement:

```
mysql> INSERT INTO some_users (name,address,gmt_create,gmt_modified,intro) VALUES ('sun',SAMPLE_SEQ.n
extVal,now(),now(),'aa');
Query OK, 1 row affected (0.01 sec)
```

Note If you set the AUTO_INCREMENT parameter when creating a table, you do not need to specify an auto-increment column when running the INSERT statement. The auto-increment column is automatically maintained by DRDS.

Acquire the values of sequences in batches

Syntax:

```
SELECT < sequence name >.NEXTVAL FROM DUAL WHERE COUNT = < numeric value >
```

Example:

```
SELECT sample_seq.nextVal FROM dual WHERE count = 10;
+--------------------+
| SAMPLE_SEQ.NEXTVAL |
+--------------------+
|             101002 |
|             101003 |
|             101004 |
|             101005 |
|             101006|
|             101007 |
|             101008 |
|             101009 |
|             101010 |
|             101011 |
+--------------------+
10 rows in set (0.04 sec)
```

13.1.13.3. Implicit sequence usage

After AUTO_INCREMENT is set for a primary key, the primary key is automatically filled in by using a sequence which is maintained by Distributed Relational Database Service (DRDS).

CREATE TABLE

The standard CREATE TABLE syntax is extended to add the sequence type for auto-increment columns. If the type keyword is not specified, the default type is GROUP. The sequence names that are automatically created by DRDS and associated with tables are prefixed with AUTO_SEQ_ and suffixed with the table name.

```
CREATE TABLE <name> (
   <column> ... AUTO_INCREMENT [ BY GROUP | SIMPLE | TIME ],
   <column definition>,
   ...
) ... AUTO_INCREMENT=<start value>
```

SHOW CREATE TABLE

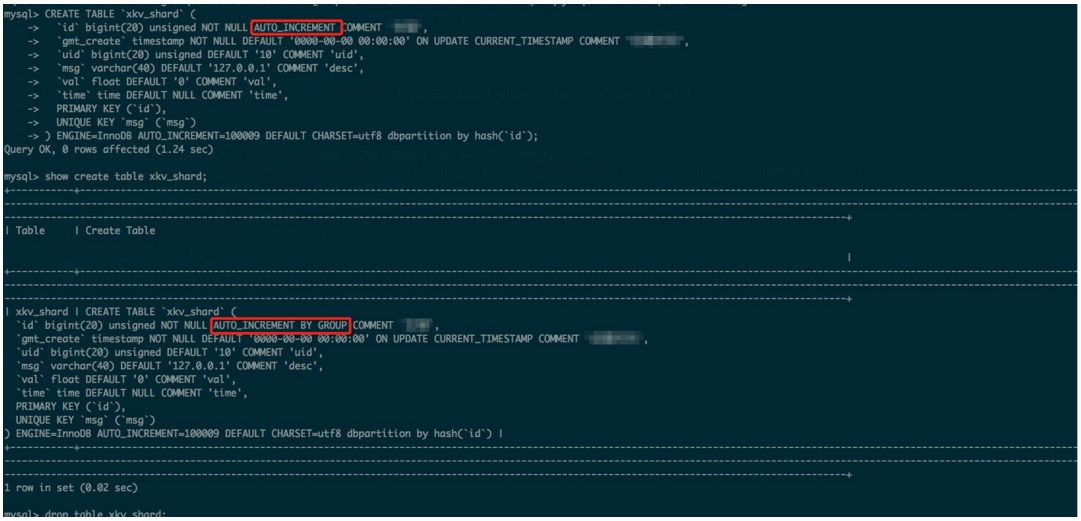
The sequence type is displayed for the auto-increment column of a table shard or broadcast table.

```
SHOW CREATE TABLE <name>
```

Examples

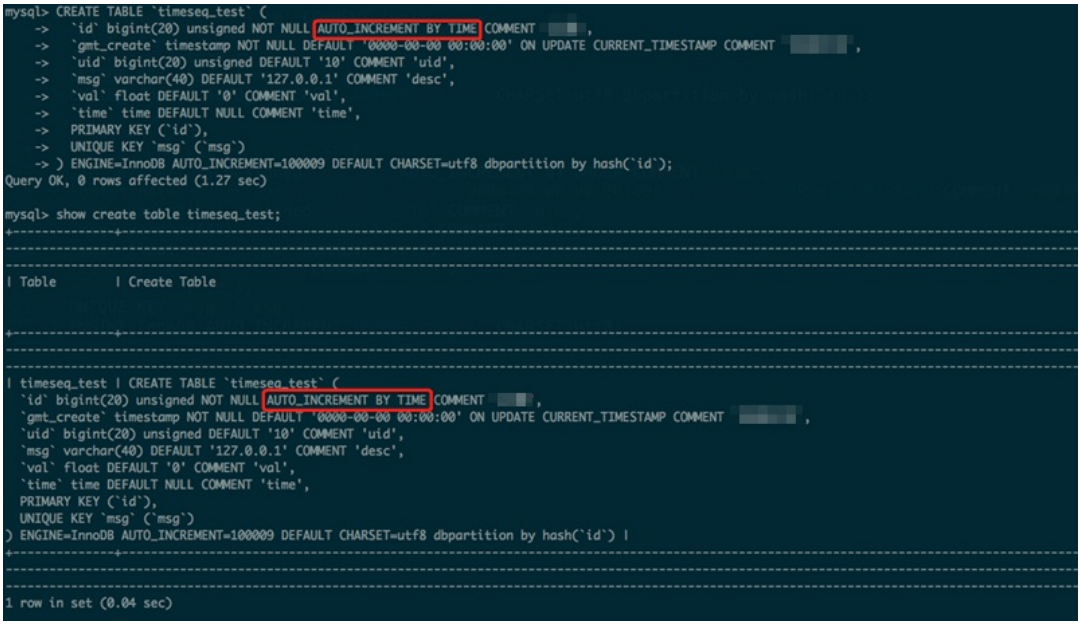
- If AUTO_INCREMENT is set but the sequence type is not specified when a table is created, the group sequence type is used by default.

  Example 1



- When creating a table, set AUTO_INCREMENT and specify a time-based sequence as the primary key value.

  Example 2



ALTER TABLE

Currently, **ALTER TABLE** cannot be used to change the sequence type but can be used to change the initial value. If you want to modify the data of the implicit sequence type in a table, run **SHOW SEQUENCES** to find the names and types of sequences and run **ALTER SEQUENCE** to modify the data.

```
ALTER TABLE <name> ... AUTO_INCREMENT=<start value>
```

Notice Exercise caution when changing the initial value of AUTO_INCREMENT after DRDS sequences are used. You need to analyze the existing sequence values and the speed of creating sequence values to avoid conflicts.

13.1.13.4. Limits and precautions for sequences

This topic describes the limits and precautions for sequences.

Limits and precautions

- When a time-based sequence is used in the auto-increment column of a table, the column must be of the BIGINT type.
- `START WITH` must be set when the sequence is changed to another type.
- When the `INSERT` statement is executed on a database in non-partition mode where only one ApsaraDB RDS for MySQL database is bound or on a database in partition mode that has only one table but no broadcast table, automatically optimizes and sends the statement, and bypasses the part of the optimizer that allocates the sequence value. In this case, `INSERT INTO ... VALUES (seq.nextval, ...)` is not supported. We recommend that you use the ApsaraDB RDS for MySQL auto-increment column feature instead.
- If the hint for a specific database shard is used by the `INSERT` statement such as `INSERT INTO ... VALUES ...` or `INSERT INTO … SELECT…` and the target table uses a sequence, bypasses the optimizer and directly sends the statement so that the sequence does not take effect. The target table creates an ID by using the auto-increment feature of the ApsaraDB RDS for MySQL table.
- The auto-increment ID allocation method for the same table must be kept consistent, which may be based on sequences or the auto-increment column of the ApsaraDB RDS for MySQL. If both of the two allocation methods are used for the same table, duplicate IDs may be created and making location difficult.

Troubleshoot primary key conflicts

Assume that data is directly written to ApsaraDB RDS for MySQL and that the related primary key value is not the sequence value created by . If automatically creates a primary key and writes it to the database, this primary key may conflict with that of the directly written data. This problem can be resolved as follows:

1. View the existing sequences by using the -specified SQL statement. The sequence prefixed with `AUTO_SEQ_` is an implicit sequence. This sequence is generated when a table is created with the `AUTO_INCREMENT` parameter.

```
mysql> SHOW SEQUENCES;
+-------------------------+-------+--------------+------------+-----------+-------+-------+
| NAME                    | VALUE | INCREMENT_BY | START_WITH | MAX_VALUE | CYCLE | TYPE  |
+-------------------------+-------+--------------+------------+-----------+-------+-------+
| AUTO_SEQ_timeseq_test   | N/A   | N/A          | N/A        | N/A       | N/A   | TIME  |
| AUTO_SEQ_xkv_shard_tbl1 | 0     | N/A          | N/A        | N/A       | N/A   | GROUP |
| AUTO_SEQ_xkv_shard      | 0     | N/A          | N/A        | N/A       | N/A   | GROUP |
+-------------------------+-------+--------------+------------+-----------+-------+-------+
3 rows in set (0.04 sec)
```

2. For example, if the t_item table contains conflicts and its primary key is ID, then retrieve the maximum primary key value of this table from :

```
mysql> SELECT MAX(id) FROM t_item;
+-----------+
| max(id)   |
+-----------+
| 8231 |
+-----------+
1 row in set (0.01 sec)
```

3. Update the related value in the sequence table to a value greater than 8231, such as 9000. Then, no error is returned for the auto-increment primary key created by the subsequent INSERT statement.

```
mysql> ALTER SEQUENCE AUTO_SEQ_USERS START WITH 9000;
Query OK, 1 row affected (0.01 sec)
```

13.1.14. Best practices

13.1.14.1. Select a shard key

A shard key is a field for database sharding and table sharding, which is used to create sharding rules during horizontal partitioning. partitions a logical table horizontally into the physical database shards on each ApsaraDB RDS for MySQL instance based on the shard key.

The primary principle of sharding is to identify the business logic-specific subject of data in a table as much as possible and confirm that most (or core) database operations are performed based on this subject. Then, use the subject-related field as the shard key to perform database sharding and table sharding.

The business logic-specific subject is related to business scenarios. The following typical scenarios include business logic-specific subjects that can be used as shard keys:

- User-oriented Internet applications are operated to meet user requirements. Users are the business logic-specific subject and the user-related field can be used as the shard key.
- Seller-oriented e-commerce applications are operated to meet seller requirements. Sellers are the business logic-specific subject and the seller-related field can be used as the shard key.
- Game-oriented applications are operated to meet gamer requirements. Gamers are the business logic-specific subject and the gamer-related field can be used as the shard key.
- Internet of Vehicles (IoV) applications are operated based on vehicle information. Vehicles are the business logic-specific subject and the vehicle-related field can be used as the shard key.
- Tax-oriented applications are operated based on taxpayer information to provide frontend services. Taxpayers are the business logic-specific subject and the taxpayer-related field can be used as the shard key.

In other scenarios, you can also use the appropriate subject of business logic as the shard key.

For example, in a seller-oriented e-commerce application, the following single table must be horizontally partitioned:

```
CREATE TABLE sample_order (
  id INT(11) NOT NULL,
  sellerId INT(11) NOT NULL,
  trade_id INT(11) NOT NULL,
  buyer_id INT(11) NOT NULL,
  buyer_nick VARCHAR(64) DEFAULT NULL,
  PRIMARY KEY (id)
)
```

The sellerId field is used as the shard key because seller is the business logic-specific subject. In the case of database sharding but no table sharding, the distributed data definition language (DDL) statement for table creation is as follows:

```
CREATE TABLE sample_order (
  id INT(11) NOT NULL,
  sellerId INT(11) NOT NULL,
  trade_id INT(11) NOT NULL,
  buyer_id INT(11) NOT NULL,
  buyer_nick VARCHAR(64) DEFAULT NULL,
  PRIMARY KEY (id)
) DBPARTITION BY HASH(sellerId)
```

If no business logic-specific subject can be used as the shard key, use the following methods to select an appropriate shard key:

- Determine the shard key based on the distribution and access of data. Distribute the data in a table to different physical database shards and table shards as evenly as possible. This method is applicable to scenarios with massive analytical queries, in which query concurrency stays at 1.
- Determine the shard key for database sharding and table sharding by combining fields of the numeric (string)

type and time type. This method is applicable to log retrieval.

For example, a log system records all user operations and needs to horizontally partition the following single log table:

```
CREATE TABLE user_log (
  userId INT(11) NOT NULL,
  name VARCHAR(64) NOT NULL,
  operation VARCHAR(128) DEFAULT NULL,
  actionDate DATE DEFAULT NULL
)
```

You can combine the user identifier with the time field to create a shard key for partitioning the table by week. The distributed DDL statement for table creation is as follows:

```
CREATE TABLE user_log (
  userId INT(11) NOT NULL,
  name VARCHAR(64) NOT NULL,
  operation VARCHAR(128) DEFAULT NULL,
  actionDate DATE DEFAULT NULL
) DBPARTITION BY HASH(userId) TBPARTITION BY WEEK(actionDate) TBPARTITIONS 7
```

For more information about shard key selection and table shard forms, see DDL statements.

Notice Avoid using hotspot data as the shard key if possible.

13.1.14.2. Select the number of shards

Distributed Relational Database Service (DRDS) supports horizontal partitioning of databases and tables. By default, eight physical database shards are created on an ApsaraDB for RDS instance, and one or more physical table shards can be created on each physical database shard. The number of table shards is also called the number of shards.

Generally, we recommend that each physical table shard contain no more than 5 million rows of data. Generally, you can estimate the data growth in one to two years. Divide the estimated total data size by the total number of physical database shards, and then divide the result by the recommended maximum data size of 5 million, to obtain the number of physical table shards to be created on each physical database shard:

```
Number of physical table shards in each physical database shard = CEILING(Estimated total data size/(
Number of ApsaraDB for RDS instances x 8)/5,000,000)
```

Therefore, when the calculated number of physical table shards is equal to 1, only database sharding needs to be performed, that is, a physical table shard is created in each physical database shard. If the calculation result is greater than 1, we recommend that you perform both database sharding and table sharding, that is, there are multiple physical table shards in each physical database shard.

For example, if a user estimates that the total data size of a table will be about 0.1 billion rows two years later and the user has four ApsaraDB for RDS instances, then according to the preceding formula:

```
Number of physical table shards in each physical database shard = CEILING(100,000,000/(4 x 8)/5,000,0
00) = CEILING(0.625) = 1
```

If the result is 1, only database sharding is needed, that is, one physical table shard is created in each physical database shard.

If only one ApsaraDB for RDS instance is used in the preceding example, the formula is as follows:

```
Number of physical table shards in each physical database shard = CEILING(100,000,000/(1 x 8)/5,000,000) = CEILING(2.5) = 3
```

If the result is 3, we recommend that you create three physical table shards in each physical database shard.

13.1.14.3. Basic concepts of SQL optimization

Distributed Relational Database Service (DRDS) is an efficient and stable distributed relational database service that processes distributed relational computing. DRDS optimizes SQL statements differently from single-instance relational databases such as MySQL. DRDS focuses on the network I/O overheads in a distributed environment and pushes SQL operations down to the underlying database shards (such as databases on ApsaraDB for RDS instances) for execution, thereby reducing the network I/O overheads and improving the SQL execution efficiency.

DRDS provides commands for obtaining the SQL execution information to help SQL optimization, for example, EXPLAIN commands for obtaining SQL execution plans and TRACE commands for obtaining SQL execution processes and overheads. This topic describes the basic concepts and common commands related to SQL optimization in DRDS.

Execution plan

An SQL execution plan (or execution plan) is a set of ordered operation steps generated to access data. In DRDS, the execution plan is divided into the execution plan at the DRDS layer and the execution plan at the ApsaraDB for RDS layer. Execution plan analysis is an effective way to optimize SQL statements. Through execution plan analysis, you can know whether DRDS or ApsaraDB for RDS has generated optimal execution plans for SQL statements and whether further optimization can be made.

During SQL statement execution, based on the basic information of the SQL statement and related tables, the DRDS optimizer determines on which the database shards the SQL statement should be executed, and the specific SQL statement form, execution policy, and data merging and computing policy for each database shard. This process optimizes SQL statement execution and generates execution plans at the DRDS layer. The execution plan at the ApsaraDB for RDS layer is the native MySQL execution plan.

DRDS provides a set of EXPLAIN commands to display execution plans at different levels or with different levels of detail.

The following table briefly describes the EXPLAIN commands in DRDS.

EXPLAIN command description

Command	Description	Example
EXPLAIN { SQL }	Displays the summary execution plan of SQL statements at the DRDS layer, including the database shards on which the SQL statement is run, physical statements, and general parameters.	EXPLAIN SELECT * FROM test
EXPLAIN DETAIL { SQL }	Displays the detailed execution plans of SQL statements at the DRDS layer, including the statement type, concurrency, returned field information, physical tables, and database groups.	EXPLAIN DETAIL SELECT * FROM test
EXPLAIN EXECUTE { SQL }	Displays the execution plan of the underlying ApsaraDB for RDS instance, which is equivalent to the EXPLAIN statement of MySQL.	EXPLAIN EXECUTE SELECT * FROM test

Execution plans at the DRDS layer

The following table describes the fields in the results returned for a DRDS-layer execution plan.

Description of fields in DRDS-layer execution plans

Field	Description
GROUP_NAME	The name of the DRDS database shard. The suffix identifies the specific database shard. The value is consistent with the result of the SHOW NODE command.
SQL	The SQL statement run on this database shard.
PARAMS	The SQL statement parameters used when DRDS communicates with ApsaraDB for RDS over the Prepare protocol.

The SQL field can be in two forms:

1. If an SQL statement does not contain the following parts, the execution plan is displayed as an SQL statement:
   - Aggregate function involving multiple database shards.
   - Distributed JOIN queries involving multiple shards.
   - Complex subqueries.

   Example:

```
mysql> EXPLAIN SELECT * FROM test;
+-----------------------------------------------+-----------------------------------------------+----
----+
| GROUP_NAME                                    | SQL                                           | PAR
AMS |
+-----------------------------------------------+-----------------------------------------------+----
----+
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0000_RDS | select `test`.`c1`,`test`.`c2` from `test` | {}
|
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0001_RDS | select `test`.`c1`,`test`.`c2` from `test` | {}
|
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0002_RDS | select `test`.`c1`,`test`.`c2` from `test` | {}
|
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0003_RDS | select `test`.`c1`,`test`.`c2` from `test` | {}
|
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0004_RDS | select `test`.`c1`,`test`.`c2` from `test` | {}
|
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0005_RDS | select `test`.`c1`,`test`.`c2` from `test` | {}
|
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0006_RDS | select `test`.`c1`,`test`.`c2` from `test` | {}
|
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0007_RDS | select `test`.`c1`,`test`.`c2` from `test` | {}
|
+-----------------------------------------------+-----------------------------------------------+----
----+
8 rows in set (0.04 sec)
```

   The group names displayed in the GROUP_NAME field can be found in the returned result of SHOW NODE:

```
mysql> SHOW NODE;
+----+-----------------------------------------------+------------------+-----------------+-----
---------------+--------------------+
| ID | NAME                                          | MASTER_READ_COUNT | SLAVE_READ_COUNT | MAST
ER_READ_PERCENT | SLAVE_READ_PERCENT |
+----+-----------------------------------------------+------------------+-----------------+-----
---------------+--------------------+
| 0  | TESTDB_1478746391548CDTCTESTDB_OXGJ_0000_RDS |               69 |               0 | 100%
| 0%                 |
| 1  | TESTDB_1478746391548CDTCTESTDB_OXGJ_0001_RDS |               45 |               0 | 100%
| 0%                 |
| 2  | TESTDB_1478746391548CDTCTESTDB_OXGJ_0002_RDS |               30 |               0 | 100%
| 0%                 |
| 3  | TESTDB_1478746391548CDTCTESTDB_OXGJ_0003_RDS |               29 |               0 | 100%
| 0%                 |
| 4  | TESTDB_1478746391548CDTCTESTDB_OXGJ_0004_RDS |               11 |               0 | 100%
| 0%                 |
| 5  | TESTDB_1478746391548CDTCTESTDB_OXGJ_0005_RDS |                1 |               0 | 100%
| 0%                 |
| 6  | TESTDB_1478746391548CDTCTESTDB_OXGJ_0006_RDS |                8 |               0 | 100%
| 0%                 |
| 7  | TESTDB_1478746391548CDTCTESTDB_OXGJ_0007_RDS |               50 |               0 | 100%
| 0%                 |
+----+-----------------------------------------------+------------------+-----------------+-----
---------------+--------------------+
8 rows in set (0.10 sec)
```

2. Execution plans that cannot be expressed by SQL statements can be expressed by DRDS in custom format.

   Example:

```
mysql> EXPLAIN DETAIL SELECT COUNT(*) FROM test;
+-----------------------------------------------+-----------------------------------+--------+
| GROUP_NAME                                    | SQL                               | PARAMS |
+-----------------------------------------------+-----------------------------------+--------+
| TEST_DB_1478746391548CDTCTEST_DB_OXGJ_0000_RDS | Merge as test
    queryConcurrency:GROUP_CONCURRENT
    columns:[count(*)]
    executeOn: TEST_DB_1478746391548CDTCTEST_DB_OXGJ_0000_RDS
        Query from test as test
            queryConcurrency:SEQUENTIAL
            columns:[count(*)]
            tableName:test
            executeOn: TEST_DB_1478746391548CDTCTEST_DB_OXGJ_0000_RDS
        Query from test as test
            queryConcurrency:SEQUENTIAL
            columns:[count(*)]
            tableName:test
            executeOn: TEST_DB_1478746391548CDTCTEST_DB_OXGJ_0001_RDS
        ... ...
        Query from test as test
            queryConcurrency:SEQUENTIAL
            columns:[count(*)]
            tableName:test
            executeOn: TEST_DB_1478746391548CDTCTEST_DB_OXGJ_0007_RDS
| NULL   |
+-----------------------------------------------+-----------------------------------+--------+
1 row in set (0.00 sec)
```

executeOn in the SQL statement field indicates the database shard on which the SQL statement is run. DRDS finally merges the results returned by the database shards.

Execution plans at the ApsaraDB for RDS layer

The execution plans at the ApsaraDB for RDS layer are the same as the native MySQL execution plan. For more information, see official MySQL documentation.

One DRDS logical table may consist of multiple shards distributed in different database shards. Therefore, you can view the execution plans at the ApsaraDB for RDS layer in multiple ways.

1. View the execution plan of an ApsaraDB for RDS database shard.

   If the query condition contains a shard key, directly run the EXPLAIN EXECUTE command to display the execution plan on the corresponding database shard. Example:

```
mysql> EXPLAIN EXECUTE SELECT * FROM test WHERE c1 = 1;
+----+-------------+-------+-------+---------------+---------+---------+-------+------+-------+
| id | select_type | table | type  | possible_keys | key     | key_len | ref   | rows | Extra |
+----+-------------+-------+-------+---------------+---------+---------+-------+------+-------+
|  1 | SIMPLE      | test  | const | PRIMARY       | PRIMARY | 4       | const |    1 | NULL  |
+----+-------------+-------+-------+---------------+---------+---------+-------+------+-------+
1 row in set (0.04 sec)
```

   Notice If an SQL statement involves multiple shards (for example, its condition does not contain a shard key), the EXPLAIN EXECUTE command returns an execution plan on a random ApsaraDB for RDS database shard.

   To view the execution plan of an SQL statement on a specified database shard, you can use the Hint method. Example:

```
mysql> /! TDDL:node='TESTDB_1478746391548CDTCTESTDB_OXGJ_0000_RDS'*/EXPLAIN SELECT * FROM test;
+----+-------------+-------+------+---------------+------+---------+------+------+-------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra |
+----+-------------+-------+------+---------------+------+---------+------+------+-------+
|  1 | SIMPLE      | test  | ALL  | NULL          | NULL | NULL    | NULL |    2 | NULL  |
+----+-------------+-------+------+---------------+------+---------+------+------+-------+
1 row in set (0.04 sec)
```

2. View the execution plans of all ApsaraDB for RDS database shards.

   You can run SCAN Hint to display the execution plans of SQL statements on all database shards:

```
mysql> /! TDDL:scan='test'*/EXPLAIN SELECT * FROM test;
+----+-------------+-------+------+---------------+------+---------+------+------+-------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra |
+----+-------------+-------+------+---------------+------+---------+------+------+-------+
|  1 | SIMPLE      | test  | ALL  | NULL          | NULL | NULL    | NULL |    2 | NULL  |
|  1 | SIMPLE      | test  | ALL  | NULL          | NULL | NULL    | NULL |    3 | NULL  |
+----+-------------+-------+------+---------------+------+---------+------+------+-------+
2 rows in set (0.08 sec)
```

   Notice

   i. In Hint mode, DRDS only replaces the table names in case of database or table sharding, and then directly sends the logical SQL statement to ApsaraDB for RDS for execution. It will not process the result.

   ii. Execution plans obtained by using an EXPLAIN command are generated by static analysis and are not actually executed in databases.

TRACE command

The TRACE command in DRDS can track the SQL execution process and the overheads in each stage. It can be used together with the execution plan to facilitate SQL statement optimization.

The TRACE command contains two related commands: TRACE and SHOW TRACE, which must be used together.

13.1.14.4. SQL optimization methods

13.1.14.4.1. Overview

This topic describes the SQL optimization principles and methods for optimizing different types of SQL statements in .

Basic principles of SQL optimization

In , SQL computing that can be performed by ApsaraDB RDS for MySQL instances is called push-down computing. Push-down computing reduces data transmission, decreases overheads at the network layer and layer, and improves the execution efficiency of SQL statements.

Therefore, the basic principle for SQL statement optimization in is as follows: Push down as many computations as possible to ApsaraDB RDS for MySQL instances.

Push-down computations include:

- JOIN connections
- Filter conditions, such as `WHERE` or `HAVING` conditions
- Aggregate computing, such as `COUNT` and `GROUP BY`
- Sorting, such as `ORDER BY`
- Deduplication, such as `DISTINCT`
- Function computing, such as the `NOW()` function
- Subqueries

> **Notice** The preceding list only describes possible forms of push-down computations. It does not mean that all clauses or conditions or combinations of clauses or conditions can be pushed down for computing.

SQL statements of different types and containing different conditions can be optimized in different ways. The following uses some examples to describe how to optimize SQL statements:

- Single-table SQL optimization
  - Filter condition optimization
  - Optimization of the number of returned rows for a query
  - Grouping and sorting optimization
- JOIN query optimization
  - Optimization of push-down JOIN queries
  - Optimization of distributed JOIN queries
- Subquery optimization

13.1.14.4.2. Single-table SQL optimization

Single-table SQL optimization must follow the following principles:

- Make sure that the SQL statements contain the shard key.
- Use an equivalence condition for the shard key whenever possible.

- If the shard key is an IN condition, the number of values after IN should be as small as possible (far fewer than the number of shards, and remain unchanged as the business grows).
- If SQL statements do not contain a shard key, use only one of DISTINCT, GROUP BY, and ORDER BY in the same SQL statement.

Filter condition optimization

DRDS partitions data horizontally by the shard key. Therefore, the filter condition must contain the shard key as much as possible so that DRDS can push queries down to specific database shards based on the shard key value, to avoid scanning all tables in the DRDS instance.

For example, the shard key of the test table is c1. If the filter condition does not contain this shard key, full table scan is performed:

```
mysql> SELECT * FROM test WHERE c2 = 2;
+----+----+
| c1 | c2 |
+----+----+
|  2 |  2 |
+----+----+
1 row in set (0.05 sec)
```

The corresponding execution plan is as follows:

```
mysql> EXPLAIN SELECT * FROM test WHERE c2 = 2;
+-----------------------------------------------+-----------------------------------------------
-----------------+--------+
| GROUP_NAME                                    | SQL
| PARAMS |
+-----------------------------------------------+-----------------------------------------------
-----------------+--------+
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0004_RDS | select `test`.`c1`,`test`.`c2` from `test` where (
`test`.`c2` = 2) | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0007_RDS | select `test`.`c1`,`test`.`c2` from `test` where (
`test`.`c2` = 2) | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0005_RDS | select `test`.`c1`,`test`.`c2` from `test` where (
`test`.`c2` = 2) | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0002_RDS | select `test`.`c1`,`test`.`c2` from `test` where (
`test`.`c2` = 2) | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0003_RDS | select `test`.`c1`,`test`.`c2` from `test` where (
`test`.`c2` = 2) | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0006_RDS | select `test`.`c1`,`test`.`c2` from `test` where (
`test`.`c2` = 2) | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0000_RDS | select `test`.`c1`,`test`.`c2` from `test` where (
`test`.`c2` = 2) | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0001_RDS | select `test`.`c1`,`test`.`c2` from `test` where (
`test`.`c2` = 2) | {}     |
+-----------------------------------------------+-----------------------------------------------
-----------------+--------+
8 rows in set (0.00 sec)
```

The smaller the value range of the filter condition containing the shard key, the faster the DRDS query speed.

For example, in the query on the test table, the filter condition contains the value range of the shard key c1:

```
mysql> SELECT * FROM test WHERE c1 > 1 AND c1 < 4;
+----+----+
| c1 | c2 |
+----+----+
|  2 |  2 |
|  3 |  3 |
+----+----+
2 rows in set (0.04 sec)
```

The corresponding execution plan is as follows:

```
mysql> EXPLAIN SELECT * FROM test WHERE c1 > 1 AND c1 < 4;
+-----------------------------------------------+-----------------------------------------------------------
-----------------------------------------+--------+
| GROUP_NAME                                    | SQL
| PARAMS |
+-----------------------------------------------+-----------------------------------------------------------
-----------------------------------------+--------+
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0002_RDS | select `test`.`c1`,`test`.`c2` from `test` where (
(`test`.`c1` > 1) AND (`test`.`c1` < 4)) | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0003_RDS | select `test`.`c1`,`test`.`c2` from `test` where (
(`test`.`c1` > 1) AND (`test`.`c1` < 4)) | {}     |
+-----------------------------------------------+-----------------------------------------------------------
-----------------------------------------+--------+
2 rows in set (0.00 sec)
```

The equivalence condition is executed faster than the range condition. For example:

```
mysql> SELECT * FROM test WHERE c1 = 2;
+----+----+
| c1 | c2 |
+----+----+
|  2 |  2 |
+----+----+
1 row in set (0.03 sec)
```

The corresponding execution plan is as follows:

```
mysql> EXPLAIN SELECT * FROM test WHERE c1 = 2;
+-----------------------------------------------+-----------------------------------------------------------
-----------------+--------+
| GROUP_NAME                                    | SQL
| PARAMS |
+-----------------------------------------------+-----------------------------------------------------------
-----------------+--------+
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0002_RDS | select `test`.`c1`,`test`.`c2` from `test` where (
`test`.`c1` = 2) | {}     |
+-----------------------------------------------+-----------------------------------------------------------
-----------------+--------+
1 row in set (0.00 sec)
```

In addition, if you want to insert data into a table shard, the inserted field must contain a shard key.

For example, data inserted into the test table contains the shard key c1:

```
mysql> INSERT INTO test(c1,c2) VALUES(8,8);
Query OK, 1 row affected (0.07 sec)
```

Optimization of the number of returned rows for a query

When DRDS runs a query containing **LIMIT** [*offset*,] *row_count*, DRDS actually reads records before *offset* in order and directly discards them. In this way, when the value of *offset* is large, the query is slow even if the value of *row_count* is small. Take the following SQL statement as an example:

```
SELECT *
FROM sample_order
ORDER BY sample_order.id
LIMIT 10000, 2
```

Although only the 10000th and 10001st records are returned, it takes about 12 seconds to run the SQL statement because DRDS actually reads 10,002 records.

```
mysql> SELECT * FROM sample_order ORDER BY sample_order.id LIMIT 10000,2;
+--------------+------------+--------------+--------------+------------+
| id           | sellerId   | trade_id     | buyer_id     | buyer_nick |
+--------------+------------+--------------+--------------+------------+
| 242012755468 | 1711939506 | 242012755467 | 244148116334 | zhangsan   |
| 242012759093 | 1711939506 | 242012759092 | 244148138304 | wangwu     |
+--------------+------------+--------------+--------------+------------+
2 rows in set (11.93 sec)
```

The corresponding execution plan is as follows:

```
mysql> EXPLAIN SELECT * FROM sample_order ORDER BY sample_order.id LIMIT 10000,2;
+------------------+-------------+------------+
| GROUP_NAME       | SQL         | PARAMS     |
+------------------+-------------+------------+
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0004_RDS | select `sample_order`.`id`,`sample_order`.`sellerI
d`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order
` order by `sample_order`.`id` asc  limit 0,10002 | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0007_RDS | select `sample_order`.`id`,`sample_order`.`sellerI
d`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order
` order by `sample_order`.`id` asc  limit 0,10002 | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0005_RDS | select `sample_order`.`id`,`sample_order`.`sellerI
d`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order
` order by `sample_order`.`id` asc  limit 0,10002 | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0002_RDS | select `sample_order`.`id`,`sample_order`.`sellerI
d`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order
` order by `sample_order`.`id` asc  limit 0,10002 | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0003_RDS | select `sample_order`.`id`,`sample_order`.`sellerI
d`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order
` order by `sample_order`.`id` asc  limit 0,10002 | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0006_RDS | select `sample_order`.`id`,`sample_order`.`sellerI
d`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order
` order by `sample_order`.`id` asc  limit 0,10002 | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0000_RDS | select `sample_order`.`id`,`sample_order`.`sellerI
d`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order
` order by `sample_order`.`id` asc  limit 0,10002 | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0001_RDS | select `sample_order`.`id`,`sample_order`.`sellerI
d`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order
` order by `sample_order`.`id` asc  limit 0,10002 | {}     |
+------------------+-------------+------------+
8 rows in set (0.01 sec)
```

To optimize the preceding SQL statement, find the ID set, and use IN to match the actual records. The modified SQL query is as follows:

```
SELECT *
FROM sample_order o
WHERE o.id IN (
      SELECT id
      FROM sample_order
      ORDER BY id
      LIMIT 10000, 2 )
```

The purpose is to cache IDs in the memory first (on the premise that the number of IDs is small). If the shard key of the sample_order table is an ID, DRDS can also push down such an IN query to different database shards through rule-based calculation, avoiding full table scan and unnecessary network I/O. Check the result of the rewritten SQL query:

```
mysql> SELECT *
    -> FROM sample_order o
    -> WHERE o.id IN ( SELECT id FROM sample_order ORDER BY id LIMIT 10000,2 );
+--------------+------------+--------------+--------------+------------+
| id           | sellerId   | trade_id     | buyer_id     | buyer_nick |
+--------------+------------+--------------+--------------+------------+
| 242012755468 | 1711939506 | 242012755467 | 244148116334 | zhangsan   |
| 242012759093 | 1711939506 | 242012759092 | 244148138304 | wangwu     |
+--------------+------------+--------------+--------------+------------+
2 rows in set (1.08 sec)
```

The execution time is significantly reduced from 12 seconds to 1.08 seconds.

The corresponding execution plan is as follows:

```
mysql> EXPLAIN SELECT *
    -> FROM sample_order o
    -> WHERE o.id IN ( SELECT id FROM sample_order ORDER BY id LIMIT 10000,2 );
+-----------------------------------------------+-----------------------------------------------------
-------------------------------------------------------------------------------+---------+
| GROUP_NAME                                    | SQL
| PARAMS |
+-----------------------------------------------+-----------------------------------------------------
-------------------------------------------------------------------------------+---------+
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0002_RDS | select `o`.`id`,`o`.`sellerId`,`o`.`trade_id`,`o`.
`buyer_id`,`o`.`buyer_nick` from `sample_order` `o` where (`o`.`id` IN (10002)) | {}     |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0001_RDS | select `o`.`id`,`o`.`sellerId`,`o`.`trade_id`,`o`.
`buyer_id`,`o`.`buyer_nick` from `sample_order` `o` where (`o`.`id` IN (10001)) | {}     |
+-----------------------------------------------+-----------------------------------------------------
-------------------------------------------------------------------------------+---------+
2 rows in set (0.03 sec)
```

Grouping and sorting optimization

In DRDS, if an SQL query must use DISTINCT, GROUP BY, and ORDER BY at the same time, try to ensure that the fields after DISTINCT, GROUP BY, and ORDER BY are the same and the fields are shard keys. In this way, only a small amount of data is returned for the SQL query. This minimizes the network bandwidth consumed by distributed queries and removes the need to retrieve a large amount of data and sort the data in a temporary table, thereby maximizing the system performance.

13.1.14.4.3. JOIN query optimization

JOIN queries in DRDS are classified into push-down JOIN queries and non-push-down JOIN queries (distributed JOIN queries). The optimization policies for these two types of JOIN queries are different.

Optimize push-down JOIN queries

Push-down JOIN queries are classified into the following types:

- JOIN queries between single tables (non-partition tables).
- The tables involved in the JOIN query contain the shard key in the filter condition and use the same sharding algorithm (that is, the data calculated by the sharding algorithm is distributed to the same shard).
- Tables involved in the JOIN query use the shard key as the JOIN condition and use the same sharding algorithm.
- JOIN query between broadcast tables (or small table broadcast) and table shards.

In DRDS, optimize JOIN queries to push-down JOIN queries that can be executed on database shards.

Take a JOIN query between a broadcast table and table shards as an example. The broadcast table is used as the JOIN driving table (the left table in the JOIN query is called the driving table). The DRDS broadcast table stores the same data in each database shard. When the broadcast table is used as the JOIN driving table, the JOIN query between this broadcast table and table shards can be converted into single-database JOIN queries and combined for computing to improve query performance.

For example, a JOIN query is performed on the following three tables, among which the sample_area table is the broadcast table, and the sample_item and sample_buyer tables are table shards. The query execution time is about 15s:

```
mysql> SELECT sample_area.name
    -> FROM sample_item i JOIN sample_buyer b ON i.sellerId = b.sellerId JOIN sample_area a ON b.prov
ince = a.id
    -> WHERE a.id < 110107
    -> LIMIT 0, 10;
+------+
| name |
+------+
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
+------+
10 rows in set (14.88 sec)
```

If you adjust the JOIN query order and move the broadcast table to the farthest left as the JOIN driving table, the JOIN query is pushed down to a single database shard in the DRDS instance:

```
mysql> SELECT sample_area.name
    -> FROM sample_area a JOIN sample_buyer b ON b.province = a.id JOIN sample_item i ON i.sellerId =
b.sellerId
    -> WHERE a.id < 110107
    -> LIMIT 0, 10;
+------+
| name |
+------+
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
|  BJ  |
+------+
10 rows in set (0.04 sec)
```

The query execution time decreases from 15 seconds to 0.04 seconds, which is a significant improvement to the query performance.

> **Notice** The broadcast table achieves data consistency through the synchronization mechanism on database shards, with a latency of several seconds.

Optimize distributed JOIN queries

If a JOIN query cannot be pushed down (that is, the JOIN condition and filter condition do not contain the shard key), DRDS must complete part of the computing in the query. Such a query is a distributed JOIN query.

Tables in a distributed JOIN query are classified into two types based on the data size:

- Small table: A table that contains a small amount of data (less than 100 data records or less data than other tables) that is involved in JOIN computing after filtering.
- Large table: A table that contains a large amount of data (more than 100 data records or more data than other tables) that is involved in JOIN computing after filtering.

In most cases, Nested Loop and its derived algorithms are used in JOIN computing at the DRDS layer. If sorting is required for JOIN queries, the Sort Merge algorithm is used. When the Nested Loop algorithm is used, a smaller data size in the left table in a JOIN query indicates a smaller number of queries performed by DRDS on the right table. If the right table has indexes or contains a small amount of data, the JOIN query is even faster. Therefore, in DRDS, the left table of a distributed JOIN query is called the driving table. To optimize a distributed JOIN query, use a small table as the driving table and set as many filter conditions as possible for the driving table.

Take the following distributed JOIN query as an example. The query takes about 24 seconds:

```
mysql> SELECT t.title, t.price
    -> FROM sample_order o,
    ->      ( SELECT * FROM sample_item i WHERE i.id = 242002396687 ) t
    -> WHERE t.source_id = o.source_item_id AND o.sellerId < 1733635660;
+----------------------------------+--------+
| title                            | price  |
+----------------------------------|--------+
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
+----------------------------------+--------+
10 rows in set (23.79 sec)
```

The preceding JOIN query is an INNER JOIN query, with the actual size of the intermediate data involved in JOIN computing unknown. In this case, perform COUNT() on the o table and t table respectively to obtain the actual data size.

For the o table, o.sellerId < 1733635660 in the WHERE condition is only related to the o table. Then, extract and add it to the COUNT() condition of the o table. The following query result is returned:

```
mysql> SELECT COUNT(*) FROM sample_order o WHERE o.sellerId < 1733635660;
+----------+
| count(*) |
+----------+
|   504018 |
+----------+
1 row in set (0.10 sec)
```

The intermediate result of the o table contains about 500,000 records. Similarly, the t table is a subquery, which can be extracted directly for the COUNT() query:

```
mysql> SELECT COUNT(*) FROM sample_item i WHERE i.id = 242002396687;
+----------+
| count(*) |
+----------+
|        1 |
+----------+
1 row in set (0.01 sec)
```

The intermediate result of the t table contains only one record. Therefore, the o table is a large table and the t table is a small table. Use the small table as the driving table of the distributed JOIN query. The result of the adjusted JOIN query is as follows:

```
mysql> SELECT t.title, t.price
    -> FROM ( SELECT * FROM sample_item i WHERE i.id = 242002396687 ) t,
    ->      sample_order o
    -> WHERE  t.source_id = o.source_item_id AND o.sellerId < 1733635660;
+----------------------------------+--------+
| title                            | price  |
+----------------------------------|--------+
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
+----------------------------------+--------+
10 rows in set (0.15 sec)
```

The query execution time decreases from about 24 seconds to 0.15 seconds, with the query performance significantly improved.

13.1.14.4.4. Subquery optimization

When optimizing an SQL query that contains subqueries, push the subqueries down to database shards as much as possible to reduce the computing workload at the DRDS layer.

For this purpose, you can try two optimization methods:

- Rewrite subqueries into multi-table JOIN queries, and optimize the JOIN queries.
- Use the shard key in the JOIN condition or filter condition so that DRDS can push the query down to a specific database shard to avoid full table scan.

The following subquery is used as an example:

```
SELECT o. *
FROM sample_order o
WHERE NOT EXISTS
     (SELECT sellerId FROM sample_seller s WHERE o.sellerId = s.id)
```

Rewrite the subquery into a JOIN query query:

```
SELECT o. *
FROM sample_order o LEFT JOIN sample_seller s ON o.sellerId = s.id
WHERE s.id IS NULL
```

13.1.14.5. Select connection pools for an application

A database connection pool is used to manage database connections in a centralized manner, so as to improve application performance and reduce database loads.

- **Reuse resources**: Connections can be reused to avoid high performance overheads caused by frequent connection creation and release. Resource reuse can also improve the system stability.

- **Improve the system response efficiency**: After the connection initialization is complete, all requests can directly use the existing connections, which avoids the overheads of connection initialization and release and improves the system response efficiency.
- **Avoid connection leakage**: The connection pool forcibly revokes connections based on the preset de-allocation policy to avoid connection resource leakage.

We recommend that you use a connection pool to connect applications and databases for service operations. For Java programs, we recommend that you use the Druid connection pool.

The following is an example of standard Druid Spring configuration:

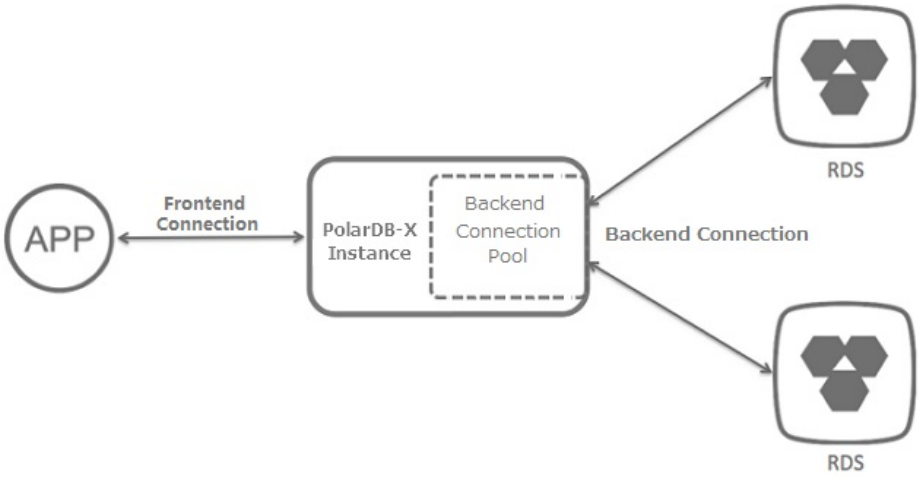
```
    <bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-m
ethod="close">
        <property name="driverClassName" value="com.mysql.jdbc.Driver" />
        <! -- Basic attributes URL, user, and password -->
        <property name="url" value="jdbc:mysql://ip:port/db? autoReconnect=true&amp;rewriteBatchedSta
tements=true&amp;socketTimeout=30000&amp;connectTimeout=3000" />
        <property name="username" value="root" />
        <property name="password" value="123456" />
        <! -- Configure the initial size, minimum value, and maximum value -->
        <property name="maxActive" value="20" />
        <property name="initialSize" value="3" />
        <property name="minIdle" value="3" />
        <! -- maxWait indicates the time-out period for obtaining the connection -->
        <property name="maxWait" value="60000" />
        <! -- timeBetweenEvictionRunsMillis indicates the interval for detecting idle connections to
be closed, in milliseconds -->
        <property name="timeBetweenEvictionRunsMillis" value="60000" />
        <! -- minEvictableIdleTimeMillis indicates the minimum idle time of a connection in the conne
ction pool, in milliseconds-->
        <property name="minEvictableIdleTimeMillis" value="300000" />
        <! -- SQL statement used to check whether connections are available -->
        <property name="validationQuery" value="SELECT 'z'" />
        <! -- Whether to enable idle connection check -->
        <property name="testWhileIdle" value="true" />
        <! -- Whether to check the connection status before obtaining a connection -->
        <property name="testOnBorrow" value="false" />
        <! -- Whether to check the connection status before releasing a connection -->
        <property name="testOnReturn" value="false" />
    </bean>
```

13.1.14.6. Connections to PolarDB-X instances

When an application connects to a instance for operation, there are two types of connections from the perspective of the instance:

- Frontend connection: a connection established by an application to the logical database in the instance.
- Backend connection: a connection established by a node in a instance to a physical database in a backend ApsaraDB RDS for MySQL instance.

  instance connection diagram



Frontend connection

Theoretically, the number of frontend connections is limited by the available memory size and the number of network connections to the nodes of the instance. However, in actual application scenarios, when an application connects to a instance, the nodes of the PolarDB-X instance usually manage a limited number of connections to perform requested operations, and do not maintain a large number of concurrent persistent connections (for example, tens of thousands of concurrent persistent connections). Therefore, the number of frontend connections that a instance can accept can be considered to be unlimited.

Considering that the number of frontend connections is unlimited and a large number of idle connections are allowed, this method applies to scenarios where a large number of servers are deployed and need to maintain their connections to the instance.

> **Note** Although the number of frontend connections is considered as unlimited, operation requests obtained from frontend connections are actually executed by internal threads of the instance through backend connections. Due to the limited number of internal threads and backend connections, the total number of concurrent requests that can be processed by the instance is limited.

Backend connection

Each node of a instance creates a backend connection pool to automatically manage and maintain the backend connections to the physical databases in the ApsaraDB RDS for MySQL instance. Therefore, the maximum number of connections in the backend connection pool of a instance is directly related to the maximum number of connections supported by the ApsaraDB RDS for MySQL instance. You can use the following formula to calculate the maximum number of connections in the backend connection pool of a instance:

```
Maximum number of connections in a backend connection pool of a  instance = FLOOR (Maximum number of connections in an ApsaraDB RDS for MySQL instance/Number of physical database shards in the ApsaraDB RDS for MySQL instance/Number of nodes on the  instance)
```

For example, a user has purchased an ApsaraDB RDS for MySQL instance and a instance of the following types:

- The ApsaraDB RDS for MySQL instance has eight physical database shards, four cores, and 16 GB memory, supporting a maximum number of 4,000 connections.
- The dedicated instance has 32 cores and 32 GB memory, with each node having two cores and 2 GB memory (that is, the instance has 16 nodes).

You can use the following formula to calculate the maximum number of connections in the backend connection pool of the instance:

```
Maximum number of connections in the backend connection pool of the  instance = FLOOR (4000/8/16) = FLOOR (31.25) = 31
```

> **Note**
>
> - The calculation result of the preceding formula is the maximum number of connections in the backend connection pool of the instance. In actual use, to reduce the connection pressure on the ApsaraDB RDS for MySQL instance, the instance adjusts the maximum number of connections in the backend connection pool to make it smaller than the upper limit.
> - We recommend that you create databases in a instance on a dedicated ApsaraDB RDS for MySQL instance. Do not create databases for other applications or instances on the dedicated ApsaraDB RDS for MySQL instance.

Relationship between frontend and backend connections

After an application establishes frontend connections to a instance and sends SQL statement execution requests, the nodes process the requests asynchronously and obtain backend connections through the internal backend connection pool, and then run optimized SQL statements on one or more physical databases.

nodes process requests asynchronously and frontend connections are not bound to backend connections. Therefore, a small number of backend connections can process a large number of requests for short transactions and simple queries from many concurrent frontend connections. This is why you need to focus on the queries per second (QPS) in , rather than the number of concurrent connections.

Although the number of frontend connections is considered to be unlimited, the maximum number of connections maintained in the backend connection pool of a instance is limited. For more information, see "Backend connections." Therefore, note the following points in actual application scenarios:

- Avoid long or large transactions in applications. These transactions occupy many or even all backend connections when they are not committed or rolled back for a long time, which reduces the overall concurrent processing capability and increases the response time (RT).
- Monitor and optimize or remove slow SQL queries run in the instance, to prevent them from occupying too many backend connections. Otherwise, the instance or the ApsaraDB RDS for MySQL instance is under greater processing pressure, which may lead to reduced concurrent processing capability, increased RT, or higher SQL execution failure rate due to execution timeout. For troubleshooting and optimization of slow SQL queries, see Troubleshoot slow SQL statements in PolarDB-X and Overview.
- Under normal use of connections and execution of queries, if the maximum number of connections in the backend connection pool of the instance is reached, contact Customer Services for assistance.

13.1.14.7. Perform instance upgrade

Database performance can be measured by the response time (RT) and queries per second (QPS). RT reflects the performance of a single SQL statement. This type of performance problem can be solved through SQL optimization. upgrade expands the capacity to improve performance, and is suitable for database access services with low latency and high QPS.

The performance of a instance depends on the performance of and ApsaraDB RDS for MySQL. Insufficient performance of any or ApsaraDB RDS for MySQL node can create a bottleneck in the overall performance. This topic describes how to observe the performance metrics of a instance and upgrade the PolarDB-X instance to solve the performance bottleneck. For more information about how to determine the performance of an ApsaraDB RDS for MySQL instance and upgrade the ApsaraDB RDS for MySQL instance, see the ApsaraDB RDS for MySQL documentation.

Determine the performance bottleneck of a instance

The QPS and CPU performance of a instance are in positive correlation. When a instance encounters a performance bottleneck, the CPU utilization of the PolarDB-X instance remains high.

Observe the CPU utilization

1. On the **Basic Information** page of the PolarDB-X instance, choose **Monitoring and Alerts > Instance Monitoring** from the left-side navigation pane.
2. On the Instance Monitoring page, select a monitoring dimension and the corresponding metrics to view details.

If the CPU utilization **exceeds 90%** or **remains above 80%**, the PolarDB-X instance faces a performance bottleneck. If there is no bottleneck for the ApsaraDB RDS for MySQL instance, the current instance specifications cannot meet the QPS performance requirements of the business. In this case, the PolarDB-X instance needs to be upgraded.

For more performance-related service monitoring scenarios and methods for configuring the CPU utilization alert.

Upgrade

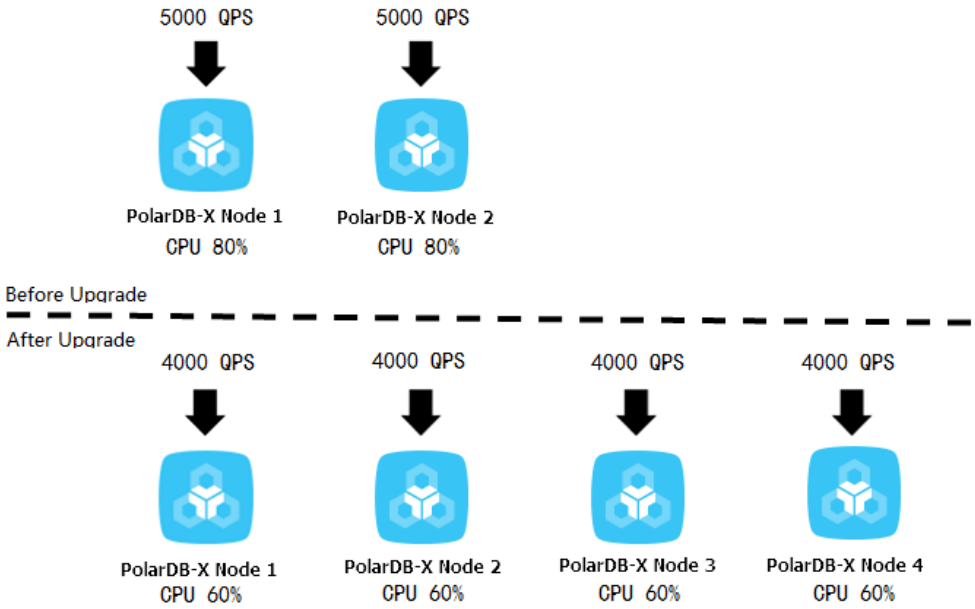
QPS is an important metric for determining whether the instance specifications can meet the business requirements. Each type of instance specifications corresponds to a reference QPS value.

Some special SQL statements require more computing (such as temporary table sorting and aggregate computing) in . In this case, the QPS supported by each instance is lower than the standard value in its type.

upgrade improves the processing performance of a PolarDB-X instance by adding nodes to share the QPS. As nodes are stateless, this upgrade method linearly improves the performance of instances.

For example, service A requires QPS of about 15 thousand. The current instance has a 4-core virtual CPU (vCPU), 4 GB memory, and two nodes, supporting QPS of only 10 thousand. After finding that the CPU utilization of the instance remains high, we upgraded the instance to 8-core vCPU and 8 GB memory, with each node handling about 4,000 QPS. Then, the performance meets service requirements, and the CPU utilization also drops to a reasonable level, as shown in the following figure.

upgrade



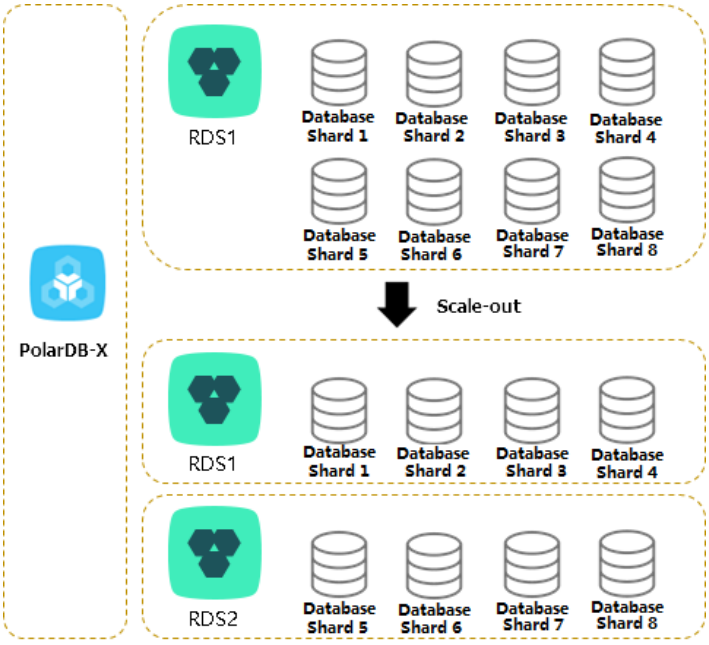
For more information about how to upgrade a PolarDB-X instance, see Change specifications.

13.1.14.8. Perform scale-out

In , smooth scale-out improves the overall performance by increasing the number of ApsaraDB RDS for MySQL instances. You can increase the number of ApsaraDB RDS for MySQL instances through smooth scale-out to increase the database capacity when the following conditions are met: 1. The input/output operations per second (IOPS), CPU utilization, disk space, and other metrics of the ApsaraDB RDS for MySQL instance reach their bottlenecks. 2. The bottlenecks cannot be removed through SQL optimization or ApsaraDB RDS for MySQL upgrade (for example, the disk has been upgraded to the top configuration).

smooth scale-out reduces the pressure on the original ApsaraDB RDS for MySQL instance by migrating database shards to the new ApsaraDB RDS for MySQL instance. For example, before scale-out, all the eight databases are deployed in one ApsaraDB RDS for MySQL instance. After scale-out, the eight databases are deployed in two ApsaraDB RDS for MySQL instances, and the pressure on a single ApsaraDB RDS for MySQL instance is significantly reduced, as shown in the following figure.

scale-out



After multiple scale-out operations, if the number of ApsaraDB RDS for MySQL instances is equal to the number of database shards, you need to create another instance and ApsaraDB RDS for MySQL databases with the expected capacity, and then migrate data to further increase the data capacity. This process is complex. We recommend that you consider the data growth expected in the next two to three years and plan the number of ApsaraDB RDS for MySQL instances properly when creating a database.

Determine whether scale-out is required

You can determine whether smooth scale-out is required based on three ApsaraDB RDS for MySQL metrics: IOPS, CPU utilization, and disk space. You can view these metrics in the ApsaraDB RDS for MySQL console. For more information, see the ApsaraDB RDS for MySQL documentation.

IOPS and CPU utilization

If you find that the IOPS or CPU utilization remains above 80% for a long time or you frequently receive alerts, follow these steps:

1. Optimize SQL statements. Generally, you can solve the high CPU utilization problem by this method.
2. If the problem persists, upgrade the ApsaraDB RDS for MySQL instance. For more information, see the ApsaraDB RDS for MySQL documentation.
3. When the CPU utilization or IOPS exceeds the threshold, you can set read-only databases to share the load on the primary database. However, read/write splitting affects read consistency. For more information, see the Read/write splitting documentation.
4. If the problem persists, scale out the instance.

Disk space

ApsaraDB RDS for MySQL has the following types of disk space:

1. Data space: the space occupied by data. The space usage continues increasing as more data is inserted. We recommend that you keep the remaining disk space above 30%.
2. System file space: the space occupied by shared tables and error log files.
3. Binary log file space: the space occupied by binary logs generated during database operation. The more update transactions there are, the larger the occupied space is.

Whether scale-out is required depends on the data space. When the data space is about to or expected to exceed the disk capacity, you can distribute the data to the databases on multiple ApsaraDB RDS for MySQL instances through scale-out.

Scale-out risks and precautions

scale-out consists of four steps: **configuration > migration > switchover > cleanup**. For more information, see the Perform smooth scale-out documentation.

Note the following points before scale-out:

- To reduce the pressure of read operations on the source ApsaraDB RDS for MySQL instance, perform scale-out when the load on the source ApsaraDB RDS for MySQL instance is low.
- During scale-out, do not submit data definition language (DDL) tasks in the console or connect to the instance to directly run DDL SQL statements. Otherwise, the scale-out task may fail.
- Scale-out requires that the source database table have a primary key. If the source database does not have a primary key, add one first.
- During scale-out, the read and write traffic is switched to the new ApsaraDB RDS for MySQL instance. The switchover process takes three to five minutes. We recommend that you perform a switchover during off-peak hours.
- Scale-out does not affect the instance before the switchover. Therefore, you can cancel the scale-out through rollback before the switchover.
- Scale-out creates pressure on databases. We recommend that you perform this operation during off-peak hours.

13.1.14.9. Troubleshoot slow SQL statements in DRDS

13.1.14.9.1. Details about a low SQL statement

defines an SQL statement that takes more than 1 second to run as a slow SQL statement. Slow SQL statements in are classified into slow logical SQL statements and slow physical SQL statements. In , an SQL statement is run step by step on and ApsaraDB RDS for MySQL nodes. Large execution loss on any node will result in slow SQL statements.

- Slow logical SQL statements are slow SQL statements sent by an application to .
- Slow physical SQL statements are slow SQL statements sent by to ApsaraDB RDS for MySQL.

Syntax

```
SHOW FULL {SLOW | PHYSICAL_SLOW} [WHERE where_condition]
                                 [ORDER BY col_name [ASC | DESC], ...]
                                 [LIMIT {[offset,] row_count | row_count OFFSET offset}]
```

Description

The `SHOW FULL SLOW` command shows slow logical SQL statements, that is, SQL statements sent by an application to .

The result set of the `SHOW FULL SLOW` command contains the following columns:

- TRACE_ID: the unique identifier of the SQL statement. A logical SQL statement and the physical SQL statements generated by the logical SQL statement have the same TRACE_ID. The TRACE_ID is also sent as a comment to ApsaraDB RDS for MySQL.
- HOST: the IP address of the client that sends the SQL statement.

  > **Notice** The client IP address may not be obtained when the network type is Virtual Private Cloud (VPC).

- START_TIME: the time when starts running the SQL statement.
- EXECUTE_TIME: the time consumed by to run the SQL statement.
- AFFECT_ROW: the number of records returned or the number of rows affected by the SQL statement.
- SQL: the statement that is run.

The `SHOW FULL PHYSICAL_SLOW` command shows the slow physical SQL statements, that is, SQL statements sent by to ApsaraDB RDS for MySQL.

The result set of `SHOW FULL PHYSICAL_SLOW` contains the following columns:

- TRACE_ID: the unique identifier of the SQL statement. A logical SQL statement and the physical SQL statements generated by the logical SQL statement have the same TRACE_ID. The TRACE_ID is also sent as a comment to ApsaraDB RDS for MySQL.
- GROUP_NAME: the name of a database group. Grouping aims to manage multiple groups of databases with identical data, such as the primary and secondary databases after data replication through ApsaraDB RDS for MySQL, which are mainly used for read/write splitting and primary/secondary switchover.
- DBKEY_NAME: the name of the database shard on which the SQL statement is run.
- START_TIME: the time when starts running the SQL statement.
- EXECUTE_TIME: the time consumed by to run the SQL statement.
- SQL_EXECUTE_TIME: the time consumed by to call ApsaraDB RDS for MySQL to run this SQL statement.
- GETLOCK_CONNECTION_TIME: the time that takes to get connections from the connection pool. If the value is large, the ApsaraDB RDS for MySQL connections have been exhausted. This is typically due to a large number of slow SQL statements. You can log on to the corresponding ApsaraDB RDS for MySQL instance and run SHOW PROCESSLIST for troubleshooting.
- CREATE_CONNECTION_TIME: the time consumed by to establish a connection to ApsaraDB RDS for MySQL. If the value is large, it is largely because the ApsaraDB RDS for MySQL instance is overloaded or faulty.
- AFFECT_ROW: the number of records returned or the number of rows affected by the SQL statement.
- SQL: the statement that is run.

Example 1

The following example describes how to locate the execution of a slow SQL statement on and between and ApsaraDB RDS for MySQL.

1. You can use certain conditions, such as the execution time and SQL string match, to obtain the specified slow SQL statement:

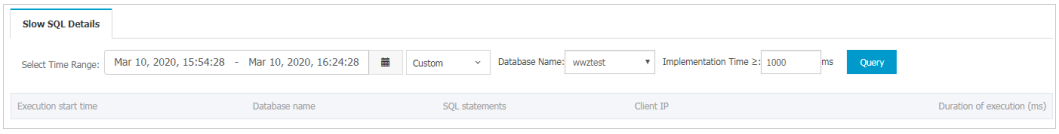
```
mysql> show full slow where `SQL` like '%select sleep(50)%';
+-----------------+-----------+-------------------------+--------------+------------+------------
------+
| TRACE_ID        | HOST      | START_TIME              | EXECUTE_TIME | AFFECT_ROW | SQL
|
+-----------------+-----------+-------------------------+--------------+------------+------------
------+
| ae0e565b8c00000 | 127.0.0.1 | 2017-03-29 19:28:43.028 |        50009 |          1 | select slee
p(50) |
+-----------------+-----------+-------------------------+--------------+------------+------------
------+
1 row in set (0.02 sec)
```

2. Based on the TRACE_ID of the slow logical SQL statement, run `SHOW FULL PHYSICAL_SLOW` to obtain the physical execution information of this SQL statement.

```
mysql> show full physical_slow where trace_id = 'ae0e565b8c00000';
+-----------------+-----------------------------------------------------------+-------------------------
---------------------+-------------------------+--------------+------------------+---------------
----------+-------------------------+------------+------------------+
| TRACE_ID        | GROUP_NAME                                                | DBKEY_NAME
| START_TIME              | EXECUTE_TIME | SQL_EXECUTE_TIME | GETLOCK_CONNECTION_TIME | CREATE_CON
NECTION_TIME | AFFECT_ROW | SQL              |
+-----------------+-----------------------------------------------------------+-------------------------
---------------------+-------------------------+--------------+------------------+---------------
----------+-------------------------+------------+------------------+
| ae0e565b8c00000 | PRIV_TEST_1489167306631PJAFPRIV_TEST_APKK_0000_RDS | rdso6g5b6206sdq832ow_pri
v_test_apkk_0000_nfup | 2017-03-29 19:27:53.02 |        50001 |            50001 |
0 |                    0 |          1 | select sleep(50) |
+-----------------+-----------------------------------------------------------+-------------------------
---------------------+-------------------------+--------------+------------------+---------------
----------+-------------------------+------------+------------------+
1 row in set (0.01 sec)
```

3. In the SQL statement details and slow SQL statement records of the ApsaraDB RDS for MySQL instance, you can query the execution information of this SQL statement on the ApsaraDB RDS for MySQL instance based on TRACE_ID.

   Slow query logs



Example 2

This example describes how to locate the original SQL statement in based on the slow SQL statement located in ApsaraDB RDS for MySQL.

1. Based on the slow SQL query log in ApsaraDB RDS for MySQL, TRACE_ID of the slow SQL statement is ae0e55660c00000.
2. Based on the TRACE_ID obtained in Step 1, run `SHOW FULL PHYSICAL_SLOW` to obtain the physical execution information of this SQL statement.

```
mysql> show full physical_slow where trace_id = 'ae0e55660c00000';
+-----------------+-----------------------------------------------------+-------------------------
----------------------+-------------------------+--------------+-------------------+--------------
-----------+-------------------------+------------+------------------+
| TRACE_ID        | GROUP_NAME                                          | DBKEY_NAME
| START_TIME              | EXECUTE_TIME | SQL_EXECUTE_TIME | GETLOCK_CONNECTION_TIME | CREATE_CO
NNECTION_TIME | AFFECT_ROW | SQL              |
+-----------------+-----------------------------------------------------+-------------------------
----------------------+-------------------------+--------------+-------------------+--------------
-----------+-------------------------+------------+------------------+
| ae0e55660c00000 | PRIV_TEST_1489167306631PJAFPRIV_TEST_APKK_0000_RDS | rdso6g5b6206sdq832ow_pri
v_test_apkk_0000_nfup | 2017-03-29 19:27:37.308 |        10003 |            10001 |
0 |                     0 |          1 | select sleep(10) |
+-----------------+-----------------------------------------------------+-------------------------
----------------------+-------------------------+--------------+-------------------+--------------
-----------+-------------------------+------------+------------------+
1 row in set (0.02 sec)
```

13.1.14.9.2. Locate slow SQL statements

Generally, you can locate a slow SQL statement in two ways: Obtain historical information about slow SQL statements from slow SQL statement records, or run `SHOW PROCESSLIST` to display the real-time execution information about slow SQL statements.

You can troubleshoot slow SQL statements as follows:

1. Locate slow SQL statements.
2. Locate nodes with performance loss.
3. Troubleshoot the performance loss.

> **Note** During troubleshooting, we recommend that you use the MySQL command line `mysql -hIP -PPORT -uUSER -pPASSWORD -c` to create the connection. Be sure to add `-c` to prevent the MySQL client from filtering out the comments (default operation) and therefore affecting the execution of HINT.

- View slow SQL statement records

  Run the following command to query top 10 slow SQL statements. This command can query logical SQL statements in . One logical SQL statement corresponds to SQL statements of one or more databases or tables of the ApsaraDB RDS for MySQL instance. For more information, see Details about a low SQL statement.

```
mysql> SHOW SLOW limit 10;
+-----------------+--------------+-------------------------+--------------+------------+----------
-----------------------------------------------------+
| TRACE_ID        | HOST         | START_TIME              | EXECUTE_TIME | AFFECT_ROW | SQL
|
+-----------------+--------------+-------------------------+--------------+------------+----------
-----------------------------------------------------+
| ac3133132801001 | xx.xxx.xx.97 | 2017-03-06 15:48:32.330 |       900392 |         -1 | select de
tail_url, sum(price) from t_item group by detail_url;  |
......
+-----------------+--------------+-------------------------+--------------+------------+----------
-----------------------------------------------------+
10 rows in set (0.01 sec)
```

- View real-time SQL execution information

If the execution of an SQL statement is slow in the current server, run **SHOW PROCESSLIST** to view the real-time SQL execution information in the current database. The value in the TIME column indicates how long the current SQL statement has been run.

```
mysql> SHOW PROCESSLIST WHERE COMMAND ! = 'Sleep';
+---------------+-----------+--------------------+-------------+---------+------------------------
-----------------------------------------------+-----------------------------------------------------
-------------------------------------------------+-----------+---------------+-----------+
| ID            | USER      | DB                 | COMMAND     | TIME    | STATE
| INFO
| ROWS_SENT | ROWS_EXAMINED | ROWS_READ |
+---------------+-----------+--------------------+-------------+---------+------------------------
-----------------------------------------------+-----------------------------------------------------
-------------------------------------------------+-----------+---------------+-----------+
| 0-0-352724126 | ifisibhk0 | test_123_wvvp_0000 | Query       |     13  | Sending data
| /*DRDS /42.120.74.88/ac47e5a72801000/ */select `t_item`.`detail_url`,SUM(`t_item`.`price`) from
`t_i |      NULL |          NULL |      NULL |
| 0-0-352864311 | cowxhthg0 | NULL               | Binlog Dump |     17  | Master has sent all bin
log to slave; waiting for binlog to be updated | NULL
|      NULL |          NULL |      NULL |
| 0-0-402714795 | ifisibhk0 | test_123_wvvp_0005 | Alter       |     114 | Sending data
| /*DRDS /42.120.74.88/ac47e5a72801000/ */ALTER TABLE `Persons` ADD `Birthday` date
|      NULL |          NULL |      NULL |
......
+---------------+-----------+--------------------+-------------+---------+------------------------
-----------------------------------------------+-----------------------------------------------------
-------------------------------------------------+-----------+---------------+-----------+
12 rows in set (0.03 sec)
```

The following describes each column:

- ID: the ID of the connection.
- USER: the user name of the database shard in which this SQL statement is run.
- DB: the specified database. If no database is specified, the value is NULL.
- COMMAND: the type of the command being executed. SLEEP indicates an idle connection. For more information about other commands, see MySQL thread information documentation.
- TIME: the elapsed execution time of the SQL statement, in seconds.
- STATE: the current execution status. For more information, see MySQL thread status documentation.
- INFO: the SQL statement being executed. The SQL statement may be too long to be displayed completely. You can derive the complete SQL statement based on information such as service parameters.

In the current example, the following slow SQL statement is identified:

```
ALTER TABLE `Persons` ADD `Birthday` date
```

13.1.14.9.3. Locate nodes with performance loss

When you locate a slow SQL statement in slow SQL statement records or real-time SQL execution information, you can run the TRACE command to trace the running time of the SQL statement in and ApsaraDB RDS for MySQL to locate the bottleneck.

The TRACE command actually runs the SQL statement, records the time consumed on all nodes, and returns the execution result. For more information about TRACE and other control commands, see Help statements.

> **Note** The TRACE command needs to maintain the context information of the connection. Some GUI clients may use connection pools, which results in command exceptions. Therefore, we recommend that you use the MySQL command line to run the TRACE command.

Run the following command for the identified slow SQL statement:

```
mysql> trace select detail_url, sum(distinct price) from t_item group by detail_url;
    +---------------+---------------+
    | detail_url    | sum(price)    |
    +---------------+---------------+
    | www.xxx.com   | 1084326800.00 |
    | www.xx1.com   | 1084326800.00 |
    | www.xx2.com   | 1084326800.00 |
    | www.xx3.com   | 1084326800.00 |
    | www.xx4.com   | 1084326800.00 |
    | www.xx5.com   | 1084326800.00 |
    ......
    +---------------+---------------+
    1 row in set (7 min 2.72 sec)
```

After the TRACE command is run, run SHOW TRACE to view the result. You can identify the bottleneck of the slow SQL statement based on the time consumption of each component.

```
 mysql> SHOW TRACE;
    +------+------------+--------------+-------------------------------------------------------------------+--
----------------------------------------------+---------------+--------------------------+------+------
----------------------------------------------------------------------------------------------------------
-----------------------------------+--------+
    | ID   | TIMESTAMP  | TYPE         | GROUP_NAME                                                        | D
BKEY_NAME                                    | TIME_COST(MS) | CONNECTION_TIME_COST(MS) | ROWS | STATE
MENT
| PARAMS |
    +------+------------+--------------+-------------------------------------------------------------------+--
----------------------------------------------+---------------+--------------------------+------+------
----------------------------------------------------------------------------------------------------------
-----------------------------------+--------+
    |    0 |      0.000 | Optimize     | DRDS                                                              | D
RDS                                          | 2             | 0.00                     |    0 | selec
t detail_url, sum(price) from t_item group by detail_url
| NULL   |
    |    1 | 423507.342 | Merge Sorted | DRDS                                                              | D
RDS                                          | 411307        | 0.00                     |    8 | Using
Merge Sorted, Order By (`t_item`.`detail_url` asc )
| NULL   |
    |    2 |      2.378 | Query        | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0003_RDS | r
dso6g5b6206sdq832ow_test_123_wvvp_0003_hbpz | 15            | 1.59                     |    1 | selec
t `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) from `t_item` group by `t_item`.`detail_url` o
rder by `t_item`.`detail_url` asc  | NULL   |
    |    3 |      2.731 | Query        | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | r
dso6g5b6206sdq832ow_test_123_wvvp_0000_hbpz | 11            | 1.78                     |    1 | selec
t `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) from `t_item` group by `t_item`.`detail_url` o
rder by `t_item`.`detail_url` asc  | NULL   |
    |    4 |      2.933 | Query        | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0004_RDS | r
dso6g5b6206sdq832ow_test_123_wvvp_0004_hbpz | 15            | 1.48                     |    1 | selec
t `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) from `t_item` group by `t_item`.`detail_url` o
rder by `t_item`.`detail_url` asc  | NULL   |
    |    5 |      3.111 | Query        | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | r
dso6g5b6206sdq832ow_test_123_wvvp_0001_hbpz | 15            | 1.56                     |    1 | selec
```

```
t `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) from `t_item` group by `t_item`.`detail_url` o
rder by `t_item`.`detail_url` asc  | NULL   |
    |    6 |      3.323 | Query        | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | r
dso6g5b6206sdq832ow_test_123_wvvp_0007_hbpz | 15           | 1.54                      |    1 | selec
t `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) from `t_item` group by `t_item`.`detail_url` o
rder by `t_item`.`detail_url` asc  | NULL   |
    |    7 |      3.496 | Query        | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0006_RDS | r
dso6g5b6206sdq832ow_test_123_wvvp_0006_hbpz | 18           | 1.30                      |    1 | selec
t `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) from `t_item` group by `t_item`.`detail_url` o
rder by `t_item`.`detail_url` asc  | NULL   |
    |    8 |      3.505 | Query        | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS | r
dso6g5b6206sdq832ow_test_123_wvvp_0005_hbpz | 423507       | 1.97                      |    1 | selec
t `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) from `t_item` group by `t_item`.`detail_url` o
rder by `t_item`.`detail_url` asc  | NULL   |
    |    9 |      3.686 | Query        | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0002_RDS | r
dso6g5b6206sdq832ow_test_123_wvvp_0002_hbpz | 14           | 1.47                      |    1 | selec
t `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) from `t_item` group by `t_item`.`detail_url` o
rder by `t_item`.`detail_url` asc  | NULL   |
    |   10 | 423807.906 | Aggregate    | DRDS                                                     | D
RDS                                      | 1413         | 0.00                      |    1 | Aggre
gate Function (SUM(`t_item`.`price`)), Group By (`t_item`.`detail_url` asc )
| NULL   |
    +------+------------+--------------+----------------------------------------------------------------+--
-----------------------------------------+---------------+--------------------------+------+------
-----------------------------------------------------------------------------------------------------
-----------------------------------+--------+
    11 rows in set (0.01 sec)
```

In the returned results of SHOW TRACE, you can determine which node has a long execution time based on the values (in milliseconds) in the TIME_COST column. You can also see the corresponding GROUP_NAME (that is, the or ApsaraDB RDS for MySQL node) and the STATEMENT column information (that is, the SQL statement being executed). By checking whether the value of GROUP_NAME is , you can determine whether the slow node exists in or ApsaraDB RDS for MySQL.

According to the preceding results, the Merge Sorted action on the node and the TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS node of ApsaraDB RDS for MySQL take a lot of time.

13.1.14.9.4. Troubleshoot the performance loss

Slow nodes may exist on the or ApsaraDB RDS for MySQL instance. Troubleshoot the fault accordingly after the cause is determined.

Solution for slow nodes

When the `GROUP_NAME` of a slow node is in the instance, check whether time-consuming computing operations such as Merge Sorted, Temp Table Merge, and Aggregate exist during SQL statement execution. If so, rectify it. For more information, see Overview.

Solution for slow ApsaraDB RDS for MySQL nodes

When the slow node is on the ApsaraDB RDS for MySQL instance, check the execution plan of this SQL statement on the ApsaraDB RDS for MySQL instance.

In , you can run `/! TDDL:node={GROUP_NAME}*/ EXPLAIN` to check the SQL execution plan of an ApsaraDB RDS for MySQL instance. The execution plan displays the SQL execution process information, including inter-table association and index information.

The detailed process is as follows:

1. Based on `GROUP_NAME`, assemble the HINT: `/! TDDL:node='TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS'*/`.
2. Combine the assembled HINT and the statement prefixed by EXPLAIN to form a new SQL statement and run it. The EXPLAIN command does not actually run. It only displays the execution plan of the SQL statement.

The following example describes how to query the execution plan of the identified slow node.

```
    mysql> /! TDDL:node='TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS'*/ EXPLAIN  select
`t_item`.`detail_url`,SUM(distinct `t_item`.`price`) from `t_item` group by `t_item`.`detail_url` ord
er by `t_item`.`detail_url` asc;
    +----+-------------+--------+------+---------------+-----+---------+-----+---------+-----------
--------------------+
    | id | select_type | table  | type | possible_keys | key | key_len | ref | rows    | Extra
|
    +----+-------------+--------+------+---------------+-----+---------+-----+---------+-----------
--------------------+
    |  1 | SIMPLE      | t_item | ALL  | NULL          |NULL | NULL    | NULL| 1322263 | Using tempor
ary; Using filesort |
    +----+-------------+--------+------+---------------+-----+---------+-----+---------+-----------
--------------------+
    1 row in set (0.01 sec)
```

When the preceding SQL statement is run in ApsaraDB RDS for MySQL, the message `Using temporary; Using filesort` is returned. It indicates that low SQL statement execution is caused by improper use of the index. In this case, you can correct the index and run the SQL statement again.

13.1.14.10. Handle DDL exceptions

When you run any data definition language (DDL) commands of , PolarDB-X performs the corresponding DDL operation on all table shards.

Failures can be divided into two types:

1. A DDL statement fails to be executed in a database shard. DDL execution failure in any database shard may result in inconsistent table shard structures.
2. The system does not respond for a long time after a DDL statement is executed. When you perform a DDL statement on a large table, the system may make no response for a long time due to the long execution time of the DDL statement in a database shard.

Execution failures in database shards may occur for various reasons. For example, the table you want to create already exists, the column you want to add already exists, or the disk space is insufficient.

No response for a long time is generally caused by the long execution time of a DDL statement in a database shard. Taking ApsaraDB RDS for MySQL as an example, the DDL execution time depends mostly on whether the operation is an in-place (directly modifying the source table) or copy (copying data in the table) operation. An in-place operation only requires modification of metadata, while a copy operation reconstructs the whole table and also involves log and buffer operations.

To determine whether a DDL operation is an in-place or copy operation, you can view the returned value of "rows affected" after the operation is completed.

Example:

- Change the default value of a column (this operation is very fast and does not affect the table data at all):

```
Query OK, 0 rows affected (0.07 sec)
```

- Add an index (this operation takes some time, but "0 rows affected" indicates that the table data is not replicated):

```
Query OK, 0 rows affected (21.42 sec)
```

- Change the data type of column (this operation takes a long time and reconstructs all data rows in the table):

```
Query OK, 1671168 rows affected (1 min 35.54 sec)
```

Therefore, before executing a DDL operation on a large table, perform the following steps to determine whether the operation is a fast or slow operation:

1. Copy the table structure to generate a cloned table.
2. Insert some data.
3. Perform the DDL operation on the cloned table.
4. Check whether the value of "rows affected" is 0 after the operation is completed. A non-zero value means that this operation reconstructs the entire table. In this case, you need to perform this operation in off-peak hours.

Solution for failures

DDL operations distribute all SQL statements to all database shards for parallel execution. Execution failure on any database shard does not affect the execution on other database shards. In addition, provides the CHECK TABLE command to check the structure consistency of the table shards. Therefore, failed DDL operations can be performed again, and errors reported on database shards on which the operations have been executed do not affect the execution on other database shards. Make sure that all table shards ultimately have the same structure.

Procedure for handling DDL operation failures

1. Run the **CHECK TABLE** command to check the table structure. If the returned result contains only one row and the status is normal, **the table statuses are consistent**. In this case, go to Step 2. Otherwise, go to Step 3.
2. Run the **SHOW CREATE TABLE** command to check the table structure. If the displayed table structure is the same as the expected structure after the DDL statement is run, the DDL statement is run. Otherwise, go to Step 3.
3. Run the **SHOW PROCESSLIST** command to check the statuses of all SQL statements being executed. If any ongoing DDL operations are detected, wait until these operations are completed, and then perform Steps 1 and 2 to check the table structure. Otherwise, go to Step 4.
4. Perform the DDL operation again on . If the **Lock conflict** error is reported, go to Step 5. Otherwise, go to Step 3.
5. Run the **RELEASE DBLOCK** command to release the DDL operation lock, and then go to Step 4.

The procedure is as follows:

1. Check the table structure consistency

   Run the CHECK TABLE command to check the table structure. When the returned result contains only one row and the displayed status is OK, **the table structures are consistent**.

   > **Notice** If no result is returned after you run CHECK TABLE, retry by using the CLI.

```
mysql> check table `xxxx`;
+----------------------------+-------+----------+----------+
| TABLE                      | OP    | MSG_TYPE | MSG_TEXT |
+----------------------------+-------+----------+----------+
| TDDL5_APP.xxxx             | check | status   | OK       |
+----------------------------+-------+----------+----------+
1 row in set (0.05 sec)
```

2. Check the table structure

Run the SHOW CREATE TABLE command to check the table structure. If **table structures are consistent** and **correct**, the DDL statement has been run.

```
mysql> show create table `xxxx`;
+---------+-----------------------------------------------------------------------------------------------
----------------------------+
| Table   | Create Table
|
+---------+-----------------------------------------------------------------------------------------------
----------------------------+
|  xxxx   | CREATE TABLE `xxxx` (
`id` int(11) NOT NULL DEFAULT '0',
`NAME` varchar(1024) NOT NULL DEFAULT '',
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by hash(`id`) tbpartition by hash(`id`) tbpartit
ions 3                      |
+---------+-----------------------------------------------------------------------------------------------
----------------------------+
1 row in set (0.05 sec)
```

3. Check the SQL statements being executed.

   If some DDL statement executions are slow and no response is received for a long time, you can run the SHOW PROCESSLIST command to check the status of all SQL statements being executed.

```
mysql> SHOW PROCESSLIST WHERE COMMAND ! = 'Sleep';
+---------------+-----------+--------------------+-------------+---------+-----------------------
-----------------------------------------------+-----------------------------------------------
-----------------------------------------------------+-----------+---------------+-----------+
| ID            | USER      | DB                 | COMMAND     | TIME    | STATE
| INFO
| ROWS_SENT | ROWS_EXAMINED | ROWS_READ |
+---------------+-----------+--------------------+-------------+---------+-----------------------
-----------------------------------------------+-----------------------------------------------
-----------------------------------------------------+-----------+---------------+-----------+
| 0-0-352724126 | ifisibhk0 | test_123_wvvp_0000 | Query       |      15 | Sending data
| /*DRDS /xx.xxx.xx.88/ac47e5a72801000/ */select `t_item`.`detail_url`,SUM(`t_item`.`price`) from
`t_i |      NULL |          NULL |      NULL |
| 0-0-352864311 | cowxhthg0 | NULL               | Binlog Dump |      13 | Master has sent all bi
nlog to slave; waiting for binlog to be updated | NULL
|      NULL |          NULL |      NULL |
| 0-0-402714566 | ifisibhk0 | test_123_wvvp_0005 | Query       |      14 | Sending data
| /*DRDS /xx.xxx.xx.88/ac47e5a72801000/ */select `t_item`.`detail_url`,`t_item`.`price` from `t_i
|      NULL |          NULL |      NULL |
| 0-0-402714795 | ifisibhk0 | test_123_wvvp_0005 | Alter       |     114 | Sending data
| /*DRDS /xx.xxx.xx.88/ac47e5a72801000/ */ALTER TABLE `Persons` ADD `Birthday` date
|      NULL |          NULL |      NULL |
......
+---------------+-----------+--------------------+-------------+---------+-----------------------
-----------------------------------------------+-----------------------------------------------
-----------------------------------------------------+-----------+---------------+-----------+
12 rows in set (0.03 sec)
```

   The value in the TIME column indicates the number of seconds that the command has been executed. If a command execution is too slow, as shown in the figure, you can run the **KILL '0-0-402714795'** command to cancel the slow command.

> **Notice** In , one logical SQL statement corresponds to multiple statements on database shards. Therefore, you may need to kill multiple commands to stop a logical DDL statement. You can determine the logical SQL statement to which a command belongs based on the INFO column in the SHOW PROCESSLIST result set.

4. Handle the lock conflict error

   adds a database lock before performing a DDL operation and releases the lock after the operation. The KILL DDL operation may not release the lock. If you perform the DDL operation again, the following error message will be returned:

   ```
   Lock conflict , maybe last DDL is still running
   ```

   In this case, run **RELEASE DBLOCK** to release the lock. After the command is canceled and the lock is released, run the DDL statement again during off-peak hours or when the service is stopped.

Other problems

Clients cannot display the modified table structures.

To enable some clients to obtain table structures from system tables (such as COLUMNS or TABLES), creates a shadow database in database shard 0 on your ApsaraDB RDS for MySQL instance. The shadow database name must be the same as the name of your logical database. It stores all table structures and other information in the user database.

The client obtains the table structure from the system table of the shadow database. During the processing of DDL exceptions, the table structure may be modified normally in the user database but not in the shadow database due to some reasons. In this case, you need to connect to the shadow database and perform the DDL operation on the table again in the database.

> **Notice** The CHECK TABLE command does not check whether the table structure in the shadow database is consistent with that in the user database.

13.1.14.11. Efficiently scan DRDS data

Distributed Relational Database Service (DRDS) supports efficient data scanning and uses aggregate functions for statistical summary during full table scan.

The following describes common scanning scenarios:

- **Scan of table without database or table shards**: DRDS transmits the original SQL statement to the backend ApsaraDB RDS for MySQL database for execution. In this case, DRDS supports any aggregate functions.
- **Non-full table scan**: DRDS transmits the original SQL statement to each single ApsaraDB RDS for MySQL database for execution. For example, when the shard key in the WHERE clause is Equal, non-full table scan is performed. In this case, DRDS also supports any aggregate functions.
- **Full table scan**: Currently, the supported aggregate functions are COUNT, MAX, MIN, and SUM. In addition, LIKE, ORDER BY, LIMIT, and GROUP BY are also supported during full table scan.S
- **Parallel scan of all table shards**: If you need to export data from all databases, you can run the SHOW command to view the table topology and scan all table shards in parallel. For more information, see the following section.

Traverse tables by using a hint

1. Run the SHOW TOPOLOGY FROM TABLE_NAME command to obtain the table topology.

```
mysql:> SHOW TOPOLOGY FROM DRDS_USERS;
+------+------------------+--------------+
| ID   | GROUP_NAME       | TABLE_NAME   |
+------+------------------+--------------+
|    0 | DRDS_00_RDS      | drds_users   |
|    1 | DRDS_01_RDS      | drds_users   |
+------+------------------+--------------+
2 rows in set (0.06 sec)
```

By default, the non-partition table is stored in database shard 0.

2. Traverse each table for TOPOLOGY.
   i. Run the current SQL statement in database shard 0.

   ```
   /! TDDL:node='DRDS_00_RDS'*/ SELECT * FROM DRDS_USERS;
   ```

   ii. Run the current SQL statement in database shard 1.

   ```
   /! TDDL:node='DRDS_01_RDS'*/ SELECT * FROM DRDS_USERS;
   ```

Notice We recommend that you run `SHOW TOPOLOGY FROM TABLE_NAME` to obtain the latest table topology before each scan.

Parallel scans

DRDS allows you to run mysqldump to export data. However, if you want to scan data faster, you can enable multiple sessions for each table shard to scan tables in parallel.

```
mysql> SHOW TOPOLOGY FROM LJLTEST;
+------+----------------+------------+
| ID   | GROUP_NAME     | TABLE_NAME |
+------+----------------+------------+
|    0 | TDDL5_00_GROUP | ljltest_00 |
|    1 | TDDL5_00_GROUP | ljltest_01 |
|    2 | TDDL5_00_GROUP | ljltest_02 |
|    3 | TDDL5_01_GROUP | ljltest_03 |
|    4 | TDDL5_01_GROUP | ljltest_04 |
|    5 | TDDL5_01_GROUP | ljltest_05 |
|    6 | TDDL5_02_GROUP | ljltest_06 |
|    7 | TDDL5_02_GROUP | ljltest_07 |
|    8 | TDDL5_02_GROUP | ljltest_08 |
|    9 | TDDL5_03_GROUP | ljltest_09 |
|   10 | TDDL5_03_GROUP | ljltest_10 |
|   11 | TDDL5_03_GROUP | ljltest_11 |
+------+----------------+------------+
12 rows in set (0.06 sec)
```

As shown above, the table has four database shards, and each database shard has three table shards. Run the following SQL statement to operate on the table shards of the TDDL5_00_GROUP database:

```
/! TDDL:node='TDDL5_00_GROUP'*/ select * from ljltest_00;
```

Note TDDL5_00_GROUP in HINT corresponds to the GROUP_NAME column in the results of the SHOW TOPOLOGY command. In addition, the table name in the SQL statement is the table shard name.

At this time, you can enable up to 12 sessions (corresponding to 12 table shards respectively) to process data in parallel.

13.1.15. Appendix: PolarDB-X terms

This topic lists common terms of for your reference.

Term	Description	Remarks
	is a distributed database service that was independently developed by Alibaba to solve the bottlenecks of single-instance database services. is compatible with MySQL protocols and syntax. It supports automatic sharding, smooth scale-out, auto scaling, and transparent read/write splitting, and provides O&M capabilities for distributed databases throughout their entire lifecycle.	-
TDDL	Taobao Distributed Data Layer (TDDL) was developed by Alibaba and has become a preferred component for nearly 1,000 core applications of Alibaba.	-
Console	Console is designed for database administrators (DBAs) to isolate resources as required and perform operations, such as instance management, database and table management, read/write splitting configuration, smooth scale-out, monitoring data display, and IP address whitelist.	-
DRDS Manager	DRDS Manager is designed for global O&M personnel and DBAs to manage all resources and monitor the system.	-
Server	Server is the service layer of . Multiple server nodes make up a server cluster to provide distributed database services, including the read/write splitting, routed SQL execution, result merging, dynamic database configuration, and globally unique ID (GUID).	-
Load balancer	server nodes are stateless, and therefore requests can be randomly routed to any server node. The load balancer is used to complete this task. Server Load Balancer (SLB) is used for overall output by Apsara Stack. VIPServer is typically used for Alibaba middleware output.	-
Diamond	Diamond manages the configuration and storage of . It provides the configuration functions for storage, query, and notification. In , Diamond stores the source data of databases, and configuration data including the sharding rules, and switches.	-
Data Replication System	Data Replication System migrates and synchronizes data for . Its core capabilities include full data migration and incremental data synchronization. Its derived features include smooth data import, smooth scale-out, and global secondary index. Data Replication System requires the support of ZooKeeper and Rtools.	-
instance (instance)	A instance consists of multiple server nodes. A instance can contain multiple databases.	-
instance ID (instance ID)	An instance ID uniquely identifies an instance.	-
Number of nodes on a instance	The number of server nodes in a instance.	-

Term	Description	Remarks
VIP	The virtual IP addresses (VIPs) of the load balancer can be classified as: • 1. Public VIP, which is accessible from the Internet. It is used for testing. • 2. Private VIP, which is accessible only from the Alibaba Cloud internal network.	-
VPC	Virtual Private Cloud (VPC) is generally used on Alibaba Cloud.	-
Region	A region is a geographical location, such as East China. This concept is generally used for Alibaba Cloud.	-
Azone	A physical area with independent power grids and networks within one region, such as Hangzhou Zone A. This concept is generally used for Alibaba Cloud.	-
Logical SQL statement	A logical SQL statement is an SQL statement sent from an application to .	-
Physical SQL statement	A physical SQL statement is an SQL statement obtained after parses a logical SQL statement and sends it to ApsaraDB RDS for MySQL for execution.	Logical SQL statements and physical SQL statements may be the same or different. Logical and physical SQL statements may be in a one-to-one or one-to-many mapping.
QPS	The queries per second (QPS) is the average number of logical SQL statements executed by per second in a statistical period,	instead of the number of transactions. Most control statements, such as COMMIT and SET, are not counted in QPS.
RT	The response time (RT) is the average response time (in milliseconds) of logical SQL statements executed by in a statistical period. The RT of an SQL statement is calculated as follows: (Time when writes the last packet of the result set) - (Time when receives the SQL statement)	-
Physical QPS	The physical QPS is the average number of physical SQL statements that executes on ApsaraDB RDS for MySQL per second in a statistical period.	-

Term	Description	Remarks
Physical RT	The physical RT is the average response time (in milliseconds) of physical SQL statements executed by on ApsaraDB RDS for MySQL in a statistical period. The RT of a physical SQL statement is calculated as follows: (Time when receives the result set returned by ApsaraDB RDS for MySQL) - (Time when starts to obtain the connection to ApsaraDB RDS for MySQL)	This includes the time of establishing a connection to ApsaraDB RDS for MySQL or obtaining a connection from the connection pool, the network transmission time, and the time of executing the SQL statement by ApsaraDB RDS for MySQL.
Connections	The number of connections established between the application and ,	instead of the number of connections established between and ApsaraDB RDS for MySQL.
Inbound traffic	The network traffic generated when the application sends SQL statements to .	This traffic is irrelevant to the traffic used for interaction between and ApsaraDB RDS for MySQL.
Outbound traffic	The network traffic generated when sends the result set to the application.	This traffic is irrelevant to the traffic used for interaction between and ApsaraDB RDS for MySQL.
Number of active threads (ThreadRunning)	The number of threads running on a instance. This parameter can be used to indicate the load of the instance.	-
Global	The total monitoring data of all databases on a instance.	-
Memory usage	The Java Virtual Machine (JVM) memory usage of a server process.	-
Total memory usage	The memory usage of the machine where the server node is located.	This metric is available only when servers are deployed on ECS instances. Generally, this metric is used for Alibaba Cloud.

Term	Description	Remarks
CPU utilization	The CPU utilization of the machine where a server node is located.	This metric is available only when servers are deployed on ECS instances. Generally, this metric is used for Alibaba Cloud.
System load	The load of the machine where a server node is located.	This metric is available only when servers are deployed on ECS instances. Generally, this metric is used for Alibaba Cloud.
Service port	The port used by servers to provide MySQL-based services to external applications.	Generally, the port number is 3306. However, when multiple nodes (mostly physical machines) are deployed on one machine, the port number will change accordingly.
Management port	The port used by servers to provide management application program interfaces (APIs).	Generally, the port number is the service port number plus 100.
Start time	The time when servers start.	-
Running time	The continuous running time of the servers since the last startup time.	-
Total memory size	The maximum JVM memory size of a server node.	-
Memory usage	The JVM memory that is already used by the server nodes.	-
Number of nodes	Required. The number of machines. A instance is essentially a cluster, and the number of nodes refers to the number of machines in the cluster.	-
Instance type	Required. The type of the instance, including dedicated and shared instances. A dedicated instance works in the exclusive mode. A shared instance works in the multi-tenant mode, which is generally used in Alibaba Cloud.	-
Machine type	Required. The type of the machine where a PolarDB-X server node is deployed. Valid values are Auto-selected, PHY, and ECS. The inventory is divided into physical machine inventory and virtual machine inventory according to the type of machines where the PolarDB-X servers are deployed. The two types cannot be mixed because their deployment and O&M methods are different.	-
AliUid	Required. The UID of the instance. In Apsara Stack, this ID is provided by the account system in the deployment environment.	-

Term	Description	Remarks
Backend port	The backend port of the VIP. For a server node, this port is the service port of machine where the PolarDB-X server node is deployed.	-
Frontend port	The frontend port of the VIP for user access. Each VIP has a set of frontend ports and backend ports. The VIP forwards data from frontend ports to backend ports.	-
Private network/Internet	The network type of the VIP. Valid values: • Internet: the public VIP, which is accessible from the Internet. • Private network: the private VIP (including VPC VIP), which is accessible from private networks.	-
lbId	The ID of an SLB instance, which is the unique ID of VIP. A VIP is managed based on this ID.	-
Forwarding mode	The port forwarding mode of the VIP. The following modes are supported: • FNAT: This mode is recommended when the backend machine is a virtual machine or VPC needs to be supported. • NAT: This mode can be selected when the backend machine is a physical machine. Currently, this mode is only used on Alibaba Cloud. • Open FNAT: This mode is applicable only to Alibaba Cloud.	-
VPC ID	The ID of the destination VPC, that is, the VPC to be accessed.	-
VSwitch ID	The ID of the destination VSwitch, which determines the CIDR block where the VPC VIP of the instance is in.	-
APPName	The app name of the destination database. Each database has a corresponding app name for loading configurations.	-
UserName	The user name used to log on to the destination database.	-
DBName	The name of the destination database you want to log on to.	-
IP address whitelist	Only the IP addresses specified in the IP address whitelist can access the instance.	-
Read-only instance	ApsaraDB RDS for MySQL instances where physical databases reside are divided into the following two types based on whether data can be written into the instances: • Primary instance: Both read and write requests are allowed on such an instance. In Apsara Stack, ApsaraDB RDS for MySQL is supported. In Alibaba Cloud, ApsaraDB for RDS is supported. • Read-only instance: Only read requests are allowed on such an instance. In Apsara Stack, ApsaraDB RDS for MySQL is supported. In Alibaba Cloud, ApsaraDB for RDS is supported.	-

Term	Description	Remarks
Read SQL statement	A type of SQL statements used to read data, such as the SELECT statement. determines whether an SQL statement is a read-only SQL statement when it is not in a transaction. If the SQL statement is in a transaction, PolarDB-X treats it as a write SQL statement during read/write splitting.	-
Read/write splitting	If read-only ApsaraDB RDS for MySQL instances exist, you can configure in the console to allocate read SQL statements to the primary and read-only instances proportionally. automatically identifies the type of SQL statements and allocates them proportionally.	-
Smooth scale-out	On the basis of horizontal partitioning, the data distribution on ApsaraDB RDS for MySQL instances is dynamically adjusted for scale-out. Generally, scale-out is completed asynchronously without any modification to the business code.	-
Broadcast of small tables	You can synchronize the data in a single table in a database to all database shards in advance, to convert the cross-database JOIN query into a JOIN query that can be completed on physical databases.	-
Horizontal partitioning	Horizontal partitioning distributes the data rows originally stored in one table to multiple tables based on specified rules to achieve horizontal linear scaling.	-
Partition mode	This mode allows you to create multiple database shards on an ApsaraDB RDS for MySQL instance. These database shards make up a database. In this mode, all functions can be used.	-
Non-partition mode	In this mode, a database that has been created on an ApsaraDB RDS for MySQL instance is used as a database. In this mode, only read/write splitting is allowed, while other features such as database sharding and table sharding are not allowed.	-
Imported database	An existing database on the ApsaraDB RDS for MySQL instance selected for creating a database. This is a unique concept for the creation of a PolarDB-X database.	-
Read policy	The ratio of read SQL statements assigned by to the primary and read-only ApsaraDB RDS for MySQL instances.	-
Full table scan	If no shard field is specified in a SQL statement, runs the SQL statement on all table shards and summarizes the results. You can disable this function because of its high overheads.	-
Shard key	A column in a logical table. routes data and SQL statements to a physical table based on this column.	-
Data import	The operation of importing data from an existing ApsaraDB RDS for MySQL instance to a database.	-
Full data migration	The operation of migrating all existing records from a database to . An offset is recorded before full migration starts.	-
Offset	In a MySQL binary log file, each row represents a data change operation. The position of a line in the binary log file is called an offset.	-

Term	Description	Remarks
Incremental data migration	The operation of reading all MySQL binary log records from the recorded offset, converting them into SQL statements, and then running them in . Incremental migration continues before the switchover.	-
Switchover	A step of data import and smooth scale-out, which writes all the remaining incremental records from MySQL binary logs to .	-
Cleanup	The last step of smooth scale-out, which cleans redundant data and configurations generated during smooth scale-out.	-
Heterogeneous indexing	For table shards of a database, the WHERE condition of a SQL statement for query must contain the shard key whenever possible. In this way, routes the query request to a specific database shard, improving the query efficiency. If the WHERE condition of the SQL statement does not contain the shard key, performs a full table scan. provides heterogeneous indexing to solve this problem. The data in a database shard or table shard of a instance is fully or partially synchronized to another table based on different shard keys. The destination table to which the data is synchronized is called a heterogeneous index table.	-
PolarDB-X sequence	A sequence (a 64-digit number of the BIGINT data type in MySQL) aims to ensure that the data (for example, PRIMARY KEY and UNIQUE KEY) in the defined unique field is globally unique and in ordered increments.	-
hint	To facilitate usage, defines some hints to specify special actions.	-

14.AnalyticDB for PostgreSQL

14.1. User Guide

14.1.1. What is AnalyticDB for PostgreSQL?

AnalyticDB for PostgreSQL (formerly known as HybridDB for PostgreSQL) is a distributed cloud database service that uses multiple compute nodes to provide massively parallel processing (MPP) data warehousing services.

AnalyticDB for PostgreSQL is developed based on the open source Greenplum database project and enhanced by Alibaba Cloud. This service has the following features:

- Compatible with Greenplum and all tools that support it.
- Supports OSS, JSON, and HyperLogLog, a probability cardinality estimation algorithm.
- Supports SQL:2003-compliant syntax and OLAP aggregate functions to provide flexible hybrid analysis.
- Supports both row store and column store to enhance analytics performance.
- Leverages data compression technologies to reduce storage costs.
- Provides online expansion and performance monitoring services to enable DBAs, developers, and data analysts to focus on improving enterprise productivity and creating core business value instead of managing and maintaining large numbers of MPP clusters.

14.1.2. Quick start

14.1.2.1. Overview

This topic provides a quick start guide about how to perform management tasks for AnalyticDB for PostgreSQL instances such as creating an instance and logging on to a database.

- Log on to the AnalyticDB for PostgreSQL console

  This topic describes how to log on to the AnalyticDB for PostgreSQL console.

- Create an instance

  You can create an instance in the console and then manage the instance.

- Configure a whitelist

  To ensure a secure and stable database, you must add IP addresses or CIDR blocks that are allowed to access the database to a whitelist of the instance before you use the AnalyticDB for PostgreSQL instance.

- Create an initial account

  After you create an instance, you must create an initial account to log on to the database.

- Connect to a database

  You can use a client that supports PostgreSQL or Greenplum to connect to the database.

14.1.2.2. Log on to the AnalyticDB for PostgreSQL console

This topic describes how to log on to the AnalyticDB for PostgreSQL console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

1. In the address bar, enter the URL used to log on to the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that are used to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Login** to go to the Apsara Uni-manager Management Console.

14.1.2.3. Create an instance

You can create an instance in the console and then manage the instance.

1. Log on to the AnalyticDB for PostgreSQL console.
2. In the upper-right corner of the page, click **Create Instance**.
3. On the **AnalyticDB for PostgreSQL** buy page, configure the following parameters.

Section	Parameter	Description
Region	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
	Region	The region of the instance. **Note** If you need to access the AnalyticDB for PostgreSQL instance from an ECS instance over VPC, you must deploy the instance in the same region and zone as those of the ECS instance.
	Zone	The zone of the instance.
Basic Settings	Engine	Currently, only the integrated computing and storage version is supported.
	Engine Version	The engine version of the instance.
	Node Type	The unit of computing resources. Different group types have different storage capacities and computing capabilities.
	Nodes	The number of compute nodes. An instance must contain at least two compute nodes. The performance of an instance scales linearly with the number of compute nodes.

Section	Parameter	Description
Network	Network Type	Valid values: ◦ *Classic Network*: Cloud services on a classic network are not isolated from each other. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ◦ *VPC*: A VPC helps you to build an isolated network environment in Alibaba Cloud. You can customize the route table, IP address range, and gateway in a VPC. We recommend that you select VPC for enhanced security. You can create a VPC in advance, or change the network type to VPC after instance creation.
	VPC	The VPC where the AnalyticDB for PostgreSQL instance is located. **Note** Virtual Private Cloud (VPC): You can use a VPC to build an isolated network environment in Alibaba Cloud. You can customize the route table, IP address range, and gateway in a VPC.
	VSwitch	The VSwitch where the AnalyticDB for PostgreSQL instance is located.
	IP Whitelist	The IP addresses that are allowed to access the instance.

4. After you have configured the preceding parameters, click **Submit**.

14.1.2.4. Configure a whitelist

To ensure a secure and stable database, you must add IP addresses or CIDR blocks that are allowed to access the database to a whitelist.

1. Log on to the AnalyticDB for PostgreSQL console.
2. Find the target instance and click its ID. The **Basic Information** page appears.
3. In the left-side navigation pane, click **Security Controls**. The **Security Controls** page appears.
4. On the **Whitelist Settings** tab, click **Modify** corresponding to the *default* whitelist. The **Modify Group** page appears.

   Note You can also click Clear corresponding to the *default* whitelist to delete the IP addresses of the default whitelist, and then click Add Group to create a new whitelist.

5. Delete 127.0.0.1 from the *default* whitelist and enter your IP addresses in the whitelist. The following table lists the parameters.

Parameter	Description
Whitelist Name	Specify the name of the whitelist. The whitelist name must be 2 to 32 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter and end with a letter or digit. The default whitelist cannot be modified or deleted.

Parameter	Description
IP Addresses	Enter the CIDR blocks or IP addresses that are allowed to access the database. Use commas (,) to separate multiple CIDR blocks or IP addresses. ○ A whitelist can contain IP addresses such as 10.10.10.1 and CIDR blocks such as 10.10.10.0/24. This CIDR block indicates that any IP addresses in the 10.10.10.X format have access to the database. ○ The percent sign (%) or 0.0.0.0/0 indicates that any IP addresses are allowed to access the database. **Notice** This configuration is not recommended because it reduces the security of the database. ○ Default whitelists of new instances contain the loopback address 127.0.0.1. This configuration allows no access from external IP addresses. ○ You can add up to 999 IP addresses or CIDR blocks to a whitelist group.

6. Click **OK** to create a whitelist.

What's next

- We recommend that you regularly maintain the whitelist to ensure secure access for AnalyticDB for PostgreSQL.
- You can click **Modify** or **Delete** to modify or delete custom whitelists.

14.1.2.5. Create an initial account

After you create an instance, you must create an initial account to log on to the database.

1. Log on to the AnalyticDB for PostgreSQL console.
2. Find the instance that you want to manage and click its ID. The **Basic Information** page appears.
3. In the left-side navigation pane, click **Account Management**. The **Account Management** page appears.
4. In the upper-right corner of the page, click **Create Account**. The **Create Account** page appears.
5. Enter the database account and password, and click **OK**.

Parameter	Description
Account	The name of the account must be 2 to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a letter and end with a letter or digit.
New Password	The password must be 8 to 32 characters in length. It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.
Password	Enter the password again.

14.1.2.6. Obtain the client tool

The interface protocol of AnalyticDB for PostgreSQL is compatible with Greenplum Community Edition and PostgreSQL 8.2. Because of this, you can use the Greenplum or PostgreSQL client to connect to AnalyticDB for PostgreSQL.

> **Note**
>
> Apsara Stack is an isolated environment. You must deploy software installation packages to the internal environment.

Graphical client tools

AnalyticDB for PostgreSQL users can directly use client tools that support Greenplum, such as SQL Workbench, Navicat Premium, Navicat for PostgreSQL, and pgAdmin III (1.6.3).

Command-line client psql (for RHEL 6, RHEL 7, CentOS 6, and CentOS 7)

For Red Hat Enterprise Linux (RHEL) and CentOS 6 or 7, you can download the tools from the following addresses and decompress the packages to use them:

- For RHEL 6 or CentOS 6, click hybriddb_client_package_el6.
- For RHEL version 7 or CentOS version 7, click hybriddb_client_package_el7.

Command-line client psql (for other Linux systems)

The compilation methods for client tools applicable to other Linux systems are as follows:

1. Obtain the source code by using one of the following methods:
   - Obtain the git directory. You must first install the git tool.

     ```
     git clone https://github.com/greenplum-db/gpdb.git
     cd gpdb
     git checkout 5d870156
     ```

   - Download the code.

     ```
     wget https://github.com/greenplum-db/gpdb/archive/5d87015609abd330c68a5402c1267fc86cbc9e1f.zip
     unzip 5d87015609abd330c68a5402c1267fc86cbc9e1f.zip
     cd gpdb-5d87015609abd330c68a5402c1267fc86cbc9e1f
     ```

2. Use gcc and other compilers.

   ```
   ./configure
   make -j32
   make install
   ```

3. Use psql and pg_dump. The paths of the two tools are as follows:

   psql: */usr/local/pgsql/bin/psql*

   pg_dump: */usr/local/pgsql/bin/pg_dump*

Command-line client psql (for Windows and other systems)

For client tools for Windows and other systems, go to the Pivotal website to download HybridDB Client

14.1.2.7. Connect to a database

The Greenplum Database and AnalyticDB for PostgreSQL are both developed based on PostgreSQL 8.2 and fully compatible with its messaging protocol. AnalyticDB for PostgreSQL users can use tools that support the PostgreSQL 8.2 message protocol, such as libpq, JDBC, ODBC, psycopg2, and pgAdmin III.

Context

AnalyticDB for PostgreSQL provides psql, a binary program of Red Hat. For more information about the download link, see Obtain the client tool. The Greenplum official website provides an easy-to-install installation package that includes JDBC, ODBC, and libpq. For more information, see Greenplum official documentation.

> **Note**
> - Apsara Stack is an isolated environment. To access Apsara Stack, you must prepare the necessary software installation packages in advance.
> - AnalyticDB for PostgreSQL instances can only be accessed by clients deployed on ECS instances within the same region and zone.

psql

psql is a common tool used together with Greenplum, and provides a variety of command functions. Its binary files are located in the *bin* directory of Greenplum. The procedure is as follows:

1. Connect to AnalyticDB for PostgreSQL by using one of the following methods:
   - Connection string

     ```
     psql "host=yourgpdbaddress.gpdb.rds.aliyuncs.com port=3432 dbname=postgres user=gpdbaccount password=gpdbpassword"
     ```

   - Specified parameters

     ```
     psql -h yourgpdbaddress.gp.aliyun-inc.com -p 3432 -d postgres -U gpdbaccount
     ```

     Parameters:

     - -h: specifies the host address.
     - -p: specifies the port number.
     - -d: specifies the database. The default database is postgres.
     - -U: specifies the user to connect to the database.

     In psql, you can run the `psql --help` command to view more options. You can run the `\?` command to view the commands supported in psql.
2. Enter the password to go to the psql shell interface.

   ```
   postgres=>
   ```

References

- For more information about the Greenplum psql usage, visit psql.
- AnalyticDB for PostgreSQL also supports psql statements of PostgreSQL. Pay attention to the differences between Greenplum psql and PostgreSQL psql. For more information, visit PostgreSQL 8.3.23 Documentation - psql.

pgAdmin III

pgAdmin III is a PostgreSQL graphical client and can be directly used to connect to AnalyticDB for PostgreSQL. For more information, click here. For more information about other graphical clients, see Obtain the client tool.

1. Download pgAdmin III 1.6.3 or earlier versions.

   You can download pgAdmin III 1.6.3 from the PostgreSQL website. pgAdmin III 1.6.3 supports various operating systems, such as Windows, macOS, and Linux.

> **Note** AnalyticDB for PostgreSQL is compatible with PostgreSQL 8.2. Therefore, you must use pgAdmin III 1.6.3 or earlier to connect to AnalyticDB for PostgreSQL. pgAdmin 4 and later versions are not supported.

2. Choose **File > Add Server**.
3. In the New Server Registration dialog box that appears, enter the configuration information.
4. Click **OK** to connect to AnalyticDB for PostgreSQL.

JDBC

JDBC uses the interface provided by PostgreSQL. The download methods are as follows:

Click PostgreSQL JDBC Driver to download the official JDBC of PostgreSQL, and then add it to the environment variables.

The sample code is as follows:

```
import java.sql.Connection; import java.sql.DriverManager; import java.sql.ResultSet; import java.sql.SQLException; import java.sql.Statement; public class gp_conn { public static void main(String[] args) { try { Class.forName("org.postgresql.Driver"); Connection db = DriverManager.getConnection("jdbc:postgresql://mygpdbpub.gpdb.rds.aliyuncs.com:3432/postgres","mygpdb","mygpdb"); Statement st = db.createStatement(); ResultSet rs = st.executeQuery("select * from gp_segment_configuration;"); while (rs.next()) { System.out.print(rs.getString(1)); System.out.print(" | "); System.out.print(rs.getString(2)); System.out.print(" | "); System.out.print(rs.getString(3)); System.out.print(" | "); System.out.print(rs.getString(4)); System.out.print(" | "); System.out.print(rs.getString(5)); System.out.print(" | "); System.out.print(rs.getString(6)); System.out.print(" | "); System.out.print(rs.getString(7)); System.out.print(" | "); System.out.print(rs.getString(8)); System.out.print(" | "); System.out.print(rs.getString(9)); System.out.print(" | "); System.out.print(rs.getString(10)); System.out.print(" | "); System.out.println(rs.getString(11)); } rs.close(); st.close(); } catch (ClassNotFoundException e) { e.printStackTrace(); } catch (SQLException e) { e.printStackTrace(); } } }
```

Python

Python uses psycopg2 to connect to Greenplum and PostgreSQL. The procedure is as follows:

1. Install psycopg2. There are three installation methods in CentOS:
   - Method 1: Run the `yum -y install python-psycopg2` command.
   - Method 2: Run the `pip install psycopg2` command.
   - Method 3: Run the source code:

     ```
     yum install -y postgresql-devel*
     wget http://initd.org/psycopg/tarballs/PSYCOPG-2-6/psycopg2-2.6.tar.gz
     tar xf psycopg2-2.6.tar.gz
     cd psycopg2-2.6
     python setup.py build
     sudo python setup.py install
     ```
2. Run the following commands to set PYTHONPATH and reference it:

```
import psycopg2
 sql = 'select * from gp_segment_configuration;'
 conn = psycopg2.connect(database='gpdb', user='mygpdb', password='mygpdb', host='mygpdbpub.gpdb.
rds.aliyuncs.com', port=3432)
 conn.autocommit = True
 cursor = conn.cursor()
 cursor.execute(sql)
 rows = cursor.fetchall()
 for row in rows:
     print row
 conn.commit()
 conn.close()
```

A similar output is displayed:

```
(1, -1, 'p', 'p', 's', 'u', 3022, '192.168.2.158', '192.168.2.158', None, None)(6, -1, 'm', 'm',
's', 'u', 3019, '192.168.2.47', '192.168.2.47', None, None)(2, 0, 'p', 'p', 's', 'u', 3025, '192.
168.2.148', '192.168.2.148', 3525, None)(4, 0, 'm', 'm', 's', 'u', 3024, '192.168.2.158', '192.16
8.2.158', 3524, None)(3, 1, 'p', 'p', 's', 'u', 3023, '192.168.2.158', '192.168.2.158', 3523, Non
e)(5, 1, 'm', 'm', 's', 'u', 3026, '192.168.2.148', '192.168.2.148', 3526, None)
```

libpq

libpq is the C language interface to AnalyticDB for PostgreSQL. You can use the libpq library to access and manage PostgreSQL databases in a C program. You can locate its static and dynamic libraries under the lib directory.

For the example programs, visit Example Programs.

For more information about libpq, visit PostgreSQL 9.4.17 Documentation - Chapter 31. libpq - C Library.

ODBC

PostgreSQL ODBC is an open-source version based on the GNU Lesser General Public License (LGPL) protocol. You can download it from the PostgreSQL website.

1. Install the driver.

   ```
   yum install -y unixODBC.x86_64
   yum install -y postgresql-odbc.x86_64
   ```

2. View the driver configuration.

```
cat /etc/odbcinst.ini
# Example driver definitions
# Driver from the postgresql-odbc package
# Setup from the unixODBC package
[PostgreSQL]
Description = ODBC for PostgreSQL
Driver = /usr/lib/psqlodbcw.so
Setup = /usr/lib/libodbcpsqlS.so
Driver64 = /usr/lib64/psqlodbcw.so
Setup64 = /usr/lib64/libodbcpsqlS.so
FileUsage = 1
# Driver from the mysql-connector-odbc package
# Setup from the unixODBC package
[MySQL]
Description = ODBC for MySQL
Driver = /usr/lib/libmyodbc5.so
Setup = /usr/lib/libodbcmyS.so
Driver64 = /usr/lib64/libmyodbc5.so
Setup64 = /usr/lib64/libodbcmyS.so
FileUsage = 1
```

3. Configure the DSN. Replace the `****` in the following code with the corresponding connection information.

```
[mygpdb]
Description = Test to gp
Driver = PostgreSQL
Database = ****
Servername = ****.gpdb.rds.aliyuncs.com
UserName = ****
Password = ****
Port = ****
ReadOnly = 0
```

4. Test connectivity.

```
echo "select count(*) from pg_class" | isql mygpdb
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
SQL> select count(*) from pg_class
+----------------------+
| count                |
+----------------------+
| 388                  |
+----------------------+
SQLRowCount returns 1
1 rows fetched
```

5. After ODBC is connected to the instance, connect the application to ODBC. For more information, see PostgreSQL ODBC Driver and psqlODBC HOWTO - C#.

References

- Greenplum official documentation

- PostgreSQL psqlODBC
- Compiling psqlODBC on Unix
- Download ODBC connectors
- Download JDBC connectors
- The PostgreSQL JDBC Interface

14.1.3. Instances

14.1.3.1. Reset the password

If you forget the password of your database account, you can reset the password in the AnalyticDB for PostgreSQL console.

> **Note** We recommend that you change your password periodically to ensure data security.

1. Log on to the AnalyticDB for PostgreSQL console.
2. Find the target instance and click its ID. The **Basic Information** page appears.
3.
4. Click **Reset Password** in the corresponding Actions column of the account. The **Reset Account Password** page appears.
5. After you enter and confirm the new password, click **OK**.

> **Note** The password must be 8 to 32 characters in length. It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. We recommend that you do not use a previously used password.

14.1.3.2. View monitoring information

You can go to the monitoring information page in the console to view the operation status of an instance.

1. Log on to the AnalyticDB for PostgreSQL console.
2. Find the instance that you want to manage and click its ID. The **Basic Information** page appears.
3. In the left-side navigation pane, click **Monitoring and Alarms**. The **Monitoring and Alarms** page appears.

   Specify a duration of time n up to seven days in length to view the metrics for that last n period.

14.1.3.3. Switch the network type of an instance

The default network type of an instance is Virtual Private Cloud (VPC). After an instance is created, you can switch its network type between classic network and VPC as needed.

Context

AnalyticDB for PostgreSQL supports two network types: classic network and VPC. Both network types use BGP connections, and are independent of the public network of your service provider. These network types only differ in functions, and you can choose a network type based on your requirements. The two network types are applicable to different scenarios:

- Classic network: IP addresses are allocated by Alibaba Cloud. Classic networks are easy to configure and use. This network type is suitable for users who do not need to perform complex operations, or who only require short deployment cycles.

- VPC: a logically isolated private network. You can customize the network topology and IP addresses and connect through a leased line. This network type is suitable for advanced users.

> **Warning** Switching the network type will cause the database service to stop. Proceed with caution.

1. Log on to the AnalyticDB for PostgreSQL console.
2. Find the target instance and click its ID. The **Basic Information** page appears.
3. In the left-side navigation pane, click **Database Connection**. The **Database Connection** page appears.
4. In the upper-right corner of the page, click **Switch to Classic Network** or **Switch to VPC**.
5. If you click **Switch to VPC**, you must select the destination **VPC** and **VSwitch**. Click **OK**.

   > **Note** To switch the network type to VPC, a VPC and a VSwitch must exist or be created in the zone where the instance is located.

6. If you click **Switch to Classic Network**, click **OK** in the displayed message.

> **Note** After you switch the network type, it takes 3 to 30 minutes for the instance to enter the running state.

14.1.3.4. Restart an instance

To better meet your needs, AnalyticDB for PostgreSQL automatically updates the database kernel version. When you create an instance, the latest database kernel is used by default. After a new version is released, you can restart your instance to update the database kernel and use its extended features. This topic describes how to restart an instance.

> **Warning** Restarting an instance will cause the database service to stop. Proceed with caution.

1. Log on to the AnalyticDB for PostgreSQL console.
2. Find the instance that you want to manage and click its ID. The **Basic Information** page appears.
3. In the upper-right corner of the page, click **Restart Instance**.

> **Note** The restart process typically takes from 3 to 30 minutes. During the restart period, the instance cannot provide external services. We recommend that you take precautionary measures before restarting instances. After the instance has been restarted and enters the running state, you can access the database.

14.1.3.5. Import data

14.1.3.5.1. Import or export data from or to OSS in parallel

AnalyticDB for PostgreSQL can import or export data from or to OSS tables in parallel by using the OSS external table feature, gpossext. AnalyticDB for PostgreSQL also supports GZIP compression for OSS external tables to reduce file size and storage costs. gpossext can read from and write to TEXT and CSV files, even when they are compressed in GZIP packages.

- Create an OSS external table extension (oss_ext)

  To use an OSS external table, you must first create an OSS external table extension in AnalyticDB for PostgreSQL. You must create an extension for each database that you need to access.

  - Creation statement: `CREATE EXTENSION IF NOT EXISTS oss_ext;`

- Deletion statement: `DROP EXTENSION IF EXISTS oss_ext;`

- Import data in parallel
  i. Distribute data evenly among multiple OSS files for storage. We recommend that you set the number of OSS files to an integer that is the multiple of the number of compute nodes in AnalyticDB for PostgreSQL.
  ii. Create a READABLE external table in AnalyticDB for PostgreSQL.
  iii. Execute the following statement to import data in parallel:

```
INSERT INTO <destination table> SELECT * FROM <external table>
```

> **Note**
> - The data import performance depends on the OSS performance and resources of the AnalyticDB for PostgreSQL instance, such as CPU, I/O, memory, and network resources. To ensure the best import performance, we recommend that you use column store and compression when you create a table. For example, you can specify the following clause: `WITH (APPENDONLY=true, ORIENTATION=column, COMPRESSTYPE=zlib, COMPRESSLEVEL=5, BLOCKSIZE=1048576)`. For more information, see Greenplum Database official documentation on database table creation syntax.
> - We recommend that you configure OSS and AnalyticDB for PostgreSQL instances within the same region to implement the best import performance.

- Export data in parallel
  i. Create a WRITABLE external table in AnalyticDB for PostgreSQL.
  ii. Execute the following statement to export data to OSS in parallel:

```
INSERT INTO <external table> SELECT * FROM <source table>
```

- Create OSS external tables

> **Note** The syntax to create and use external tables is the same as that of Greenplum Database, except for the syntax of location-related parameters.

```
CREATE [READABLE] EXTERNAL TABLE tablename
( columnname datatype [, ...] | LIKE othertable )
LOCATION ('ossprotocol')
FORMAT 'TEXT'
            [( [HEADER]
               [DELIMITER [AS] 'delimiter' | 'OFF']
               [NULL [AS] 'null string']
               [ESCAPE [AS] 'escape' | 'OFF']
               [NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
               [FILL MISSING FIELDS] )]
          | 'CSV'
            [( [HEADER]
               [QUOTE [AS] 'quote']
               [DELIMITER [AS] 'delimiter']
               [NULL [AS] 'null string']
               [FORCE NOT NULL column [, ...]]
               [ESCAPE [AS] 'escape']
               [NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
               [FILL MISSING FIELDS] )]
[ ENCODING 'encoding' ]
[ [LOG ERRORS [INTO error_table]] SEGMENT REJECT LIMIT count
       [ROWS | PERCENT] ]
CREATE WRITABLE EXTERNAL TABLE table_name
( column_name data_type [, ...] | LIKE other_table )
LOCATION ('ossprotocol')
FORMAT 'TEXT'
               [( [DELIMITER [AS] 'delimiter']
               [NULL [AS] 'null string']
               [ESCAPE [AS] 'escape' | 'OFF'] )]
          | 'CSV'
               [([QUOTE [AS] 'quote']
               [DELIMITER [AS] 'delimiter']
               [NULL [AS] 'null string']
               [FORCE QUOTE column [, ...]] ]
               [ESCAPE [AS] 'escape'] )]
[ ENCODING 'encoding' ]
[ DISTRIBUTED BY (column, [ ... ] ) | DISTRIBUTED RANDOMLY ]
ossprotocol:
   oss://oss_endpoint prefix=prefix_name
    id=userossid key=userosskey bucket=ossbucket compressiontype=[none|gzip] async=[true|false]
ossprotocol:
   oss://oss_endpoint dir=[folder/[folder/]...]/file_name
    id=userossid key=userosskey bucket=ossbucket compressiontype=[none|gzip] async=[true|false]
ossprotocol:
   oss://oss_endpoint filepath=[folder/[folder/]...]/file_name
    id=userossid key=userosskey bucket=ossbucket compressiontype=[none|gzip] async=[true|false]
```

Parameters

Common parameters

Parameter	Description

Parameter	Description
Protocol and endpoint	It is in the `protocol name://oss_endpoint` format. The protocol name is oss. oss_endpoint is the domain name used by users to access OSS in a region. **Note** You can access the database from a VPC host by using an internal endpoint containing "internal" in the name to avoid generating public traffic.
id	The AccessKey ID of the OSS account.
key	The AccessKey secret of the OSS account.
bucket	The bucket where the data file is located. You must use OSS to create the bucket before data import.

prefix	The prefix of the path name corresponding to the data file. Prefixes are directly matched and cannot be controlled by regular expressions. The prefix, filepath, and dir parameters are mutually exclusive and only one parameter can be specified at a time.
	• If you create a READABLE external table for data import, all OSS files that contain the specified prefix will be imported.
	  ◦ If you set prefix to test/filename, the following files will be imported:
	  ▪ test/filename
	  ▪ test/filenamexxx
	  ▪ test/filename/aa
	  ▪ test/filenameyyy/aa
	  ▪ test/filenameyyy/bb/aa
	  ◦ If you set prefix to test/filename/, only the following file out of the preceding files will be imported:
	  test/filename/aa
	• if you create a WRITABLE external table for data export, each exported file will have a unique name based on this parameter.
	
	Note One or more files can be exported for each compute node. The names of exported files are in the `prefix_tablename_uuid.x` format. uuid indicates a timestamp in microseconds as an int64 value. x indicates the node ID. You can use an external table for multiple export operations. Each export operation is assigned to a uuid value. The files exported during each operation share a uuid value.

Parameter	Description
dir	The virtual folder path in OSS. The prefix, filepath, and dir parameters are mutually exclusive and only one parameter can be specified at a time. • A folder path must end with a forward slash (/) such as `test/mydir/`. • If you use this parameter when creating an external table for data import, all files under the specified virtual directory (except for its subdirectories and contained files) will be imported. Unlike filepath, dir does not require you to specify the names of files in the directory. • If this parameter is used in creating an external table for data export, all data will be exported to multiple files within the specified directory. The names of exported files are in the `filename.x` format, where x is a digit. The values of x may not be consecutive.

filepath	The file name that contains a path in OSS. The prefix, filepath, and dir parameters are mutually exclusive and only one parameter can be specified at a time. You can only specify the filepath parameter when you create a READABLE external table for data import.
	• The file name includes the file path, but not the bucket name.
	• The filename specified for data import must be in the `filename` or `filename.x` format. The values of x must be consecutive digits starting from 1.
	For example, if filepath is set to filename and OSS contains the following files, the imported files include filename, filename.1, and filename.2, but filename.4 is not imported because filename.3 does not exist.
	`filename`
	`filename.1`
	`filename.2`
	`filename.4`

Import mode parameters

Parameter	Description
async	Specifies whether to load data asynchronously. • Asynchronous data import is enabled by default. You can set async to false or f to disable asynchronous data import. • Enables the worker thread to load data from OSS to accelerate the import performance. The default import mode is asynchronous mode. • Asynchronous data import consumes more hardware resources than normal data import.
compressiontype	The compression format of the imported file. Valid values: • none: specifies to import files without compressing them. This is the default value. • gzip: specifies compress imported files in the GZIP format. Only the GZIP format is supported.
compressionlevel	The compression level of the files written to OSS. Valid values: 1 to 9. Default value: 6.

Export mode parameters

Parameter	Description
oss_flush_block_size	The size of each data block written to OSS. Valid values: 1 MB to 128 MB. Default value: 32 MB.
oss_file_max_size	The maximum size of each file written to OSS. If the limit is exceeded, subsequent data is written to another file. Valid values: 8 MB to 4000 MB. Default value: 1024 MB.
num_parallel_worker	The number of parallel compression threads for data written to OSS. Valid values: 1 to 8. Default value: 3.

Additionally, you must pay attention to the following items for the export mode:

- WRITABLE is the keyword of the external table for data export. You must specify this keyword when creating an external table.
- Only the prefix and dir parameters are supported for data export. The filepath parameter is not supported.
- You can use the DISTRIBUTED BY clause to write data from compute nodes to OSS based on the specified distribution keys.

Other common parameters

The following error-tolerance parameters can be used for data import and export:

Error-tolerance parameters

Parameter	Description
oss_connect_timeout	The connection timeout period. Unit: seconds. Default value: 10.
oss_dns_cache_timeout	The DNS timeout period. Unit: seconds. Default value: 60.
oss_speed_limit	The minimum rate tolerated. Default value: 1024 bit/s (1 Kbit/s).
oss_speed_time	The maximum amount of time tolerated. Unit: seconds. Default value: 15.

If the default values are used for the preceding parameters, a timeout will occur when the transmission rate is lower than 1 Kbit/s for 15 consecutive seconds. For more information, see **Troubleshooting** in *OSS SDK reference*.

The other parameters are compatible with the original external table syntax of Greenplum Database. For more information about the syntax, see Greenplum Database official documentation on external table syntax. These parameters include:

- FORMAT: indicates the supported file format, such as TEXT and CSV.
- ENCODING: indicates the data encoding format of a file, such as UTF-8.
- LOG ERRORS: indicates that the clause can ignore imported erroneous data and write the data to error_table. You can also use the count parameter to specify the error reporting threshold.

Examples

```
# Create a READABLE external table of OSS.
create readable external table ossexample
        (date text, time text, open float, high float,
        low float, volume int)
        location('oss://oss-cn-hangzhou.aliyuncs.com
        prefix=osstest/example id=XXX
```

```
        key=XXX bucket=testbucket compressiontype=gzip')
        FORMAT 'csv' (QUOTE '''' DELIMITER E'\t')
        ENCODING 'utf8'
        LOG ERRORS INTO my_error_rows SEGMENT REJECT LIMIT 5;
create readable external table ossexample
        (date text, time text, open float, high float,
        low float, volume int)
        location('oss://oss-cn-hangzhou.aliyuncs.com
        dir=osstest/ id=XXX
        key=XXX bucket=testbucket')
        FORMAT 'csv'
        LOG ERRORS SEGMENT REJECT LIMIT 5;
create readable external table ossexample
        (date text, time text, open float, high float,
        low float, volume int)
        location('oss://oss-cn-hangzhou.aliyuncs.com
        filepath=osstest/example.csv id=XXX
        key=XXX bucket=testbucket')
        FORMAT 'csv'
        LOG ERRORS SEGMENT REJECT LIMIT 5;
# Create a WRITABLE external table of OSS.
create WRITABLE external table ossexample_exp
        (date text, time text, open float, high float,
        low float, volume int)
        location('oss://oss-cn-hangzhou.aliyuncs.com
        prefix=osstest/exp/outfromhdb id=XXX
        key=XXX bucket=testbucket') FORMAT 'csv'
        DISTRIBUTED BY (date);
create WRITABLE external table ossexample_exp
        (date text, time text, open float, high float,
        low float, volume int)
        location('oss://oss-cn-hangzhou.aliyuncs.com
        dir=osstest/exp/ id=XXX
        key=XXX bucket=testbucket') FORMAT 'csv'
        DISTRIBUTED BY (date);
# Create a heap table named example to which you want to import data.
create table example
        (date text, time text, open float,
         high float, low float, volume int)
         DISTRIBUTED BY (date);
# Import data to the example heap table from the ossexample table in parallel.
insert into example select * from ossexample;
# Export data from example to OSS in parallel
insert into ossexample_exp select * from example;
# As shown in the following execution plan, all compute nodes are involved in the task.
# All compute nodes read data from OSS in parallel. AnalyticDB for PostgreSQL performs a redistributi
on motion operation to compute the data by using a hash algorithm, and then distributes the data to i
ts compute nodes after computing. After a compute node receives data, it performs an insert operation
to add the data to AnalyticDB for PostgreSQL.
explain insert into example select * from ossexample;
                                        QUERY PLAN
-------------------------------------------------------------------------------------------------
 Insert (slice0; segments: 4)  (rows=250000 width=92)
   ->  Redistribute Motion 4:4  (slice1; segments: 4)  (cost=0.00..11000.00 rows=250000 width=92)
         Hash Key: ossexample.date
         ->  External Scan on ossexample  (cost=0.00..11000.00 rows=250000 width=92)
(4 rows)
# As shown in the following query plan, each compute node exports local data directly to OSS without
redistributing the data.
```

```
explain insert into ossexample_exp select * from example;
                          QUERY PLAN
------------------------------------------------------------------
 Insert (slice0; segments: 3)  (rows=1 width=92)
   ->  Seq Scan on example  (cost=0.00..0.00 rows=1 width=92)
(2 rows)
```

TEXT and CSV format description

The following parameters specify the formats of files read from and written to OSS. You can specify the parameters in the external DDL parameters.

- \n: a line delimiter or line break for TEXT and CSV files.
- DELIMITER: specifies the delimiter of columns.
  - If the DELIMITER parameter is specified, the QUOTE parameter must also be specified.
  - Recommended column delimiters include commas (,), vertical bars (|), \t, and other special characters.
- QUOTE: encloses user data that contains special characters by column.
  - Strings that contain special characters will be enclosed by QUOTE to differentiate user data from the control characters.
  - To optimize the efficiency, it is unnecessary to enclose data such as integers in QUOTE characters.
  - QUOTE cannot be the same string as specified in DELIMITER. The default value of QUOTE is double quotation marks (").
  - User data that contains QUOTE characters must also contain ESCAPE characters to differentiate user data from machine code.
- ESCAPE: specifies the escape character.
  - Place an escape character before a special character that needs to be escaped to indicate that it is not a special character.
  - If ESCAPE is not specified, the default value is the same as QUOTE.
  - You can also use other characters as ESCAPE characters such as backslashes (\), which is used by MySQL.

Default control characters for TEXT and CSV files

Default control characters for TEXT and CSV files

Control character	TEXT	CSV
DELIMITER	\t (tab)	, (comma)
QUOTE	" (double quotation mark)	" (double quotation mark)
ESCAPE	N/A	Same as QUOTE
NULL	\N (backslash-N)	Empty string without quotation marks

> **Note** All control characters must be single-byte characters.

SDK troubleshooting

The following Error log information table lists the error logs generated when an error occurs during the import or export process.

Error log information

Keyword	Description
code	The HTTP status code of the error request.
error_code	The error code returned by OSS.
error_msg	The error message returned by OSS.
req_id	The UUID used to identify the request. If you require assistance in solving a problem, you can submit a ticket containing the req_id of the failed request to OSS developers.

For more information, see . You can handle timeout-related errors by using parameters related to oss_ext.

References

- Greenplum Database official documentation on external table syntax
- Greenplum Database official documentation on table creation syntax

14.1.3.5.2. Import data from MySQL

You can use the mysql2pgsql tool to migrate tables from MySQL to AnalyticDB for PostgreSQL, Greenplum Database, PostgreSQL, or PPAS.

Background information

mysql2pgsql connects a source MySQL database to a destination AnalyticDB for PostgreSQL database, queries data to be exported from the MySQL database, and then imports the data to the destination database by using the \COPY statement. The tool supports multi-thread import. Each worker thread imports a part of database tables.

To download the binary installation package of mysql2pgsql, click here.

To view instructions on source code compilation of mysql2pgsql, click here.

Procedure

1. Modify the my.cfg configuration file to configure the connection information of source and destination databases.
   i. Modify the connection information of the source MySQL database.

   > **Note** You must have the read permissions on all user tables.

   ```
   [src.mysql]
   host = "192.168.1.1"
   port = "3306"
   user = "test"
   password = "test"
   db = "test"
   encodingdir = "share"
   encoding = "utf8"
   ```

ii. Modify the connection information of the destination PostgreSQL, PPAS, or AnalyticDB for PostgreSQL database.

> **Note** You must have the write permissions on the destination table.

```
[desc.pgsql]
connect_string = "host=192.168.1.2 dbname=test port=3432  user=test password=pgsql"
```

2. Import data by using mysql2pgsql.

```
./mysql2pgsql -l <tables_list_file> -d -n -j <number of threads> -s <schema of target table>
```

Parameters

Parameter	Description

-l	Optional. Used to specify a text file that contains tables to be synchronized. If you do not specify this parameter, all the tables in the database that is specified in the configuration file will be synchronized. `<tables_list_file>` is the name of a file that contains a collection of tables to be synchronized and conditions for table queries. The content format is as follows:
	`table1 : select * from table_big where column1 < '2016-08-05'`
	`table2 :`
	`table3`
	`table4: select column1, column2 from tableX where column1 ! = 10`
	`table5: select * from table_big where column1 >= '2016-08-05'`

-d	Optional. Indicates the table creation DDL statement that creates the destination table but does not synchronize data.
-n	Optional. Must be used along with -d to specify that the table partition definition is not included in the DDL statement.
-j	Optional. Used to specify the number of threads used for data synchronization. If you do not specify this parameter, five concurrent threads will be used by default.
-s	Optional. Used to specify the schema of the destination table. Only one schema at a time can be specified by the command. If you do not specify the parameter, the data is imported into the table under the public schema.

Typical usage

Full database migration

1. Obtain the DDL statements of the corresponding destination table by running the following command:

   ```
   ./mysql2pgsql -d
   ```

2. Create a table in the destination database based on these DDL statements with the distribution key information added.
3. Run the following command to synchronize all tables:

   ```
   ./mysql2pgsql
   ```

   This command will migrate the data from all MySQL tables in the database that is specified in the configuration file to the destination database. By default, five concurrent threads are used to read and import data from involved tables.

Partial table migration

1. Create a new file tab_list.txt and enter the following content:

```
t1
t2 : select * from t2 where c1 > 138888
```

2. Run the following command to synchronize the specified t1 and t2 tables (note that for the t2 table, only data that meets the c1 > 138888 condition is migrated):

```
./mysql2pgsql -l tab_list.txt
```

14.1.3.5.3. Import data from PostgreSQL

You can use the pgsql2pgsql tool to migrate tables across AnalyticDB for PostgreSQL, Greenplum Database, PostgreSQL, and PPAS.

Context

pgsql2pgsql supports the following features:

- Full migration across PostgreSQL, PPAS, Greenplum Database, and AnalyticDB for PostgreSQL.
- Full migration and incremental migration from PostgreSQL or PPAS (version 9.4 or later) to AnalyticDB for PostgreSQL or ApsaraDB RDS for PPAS.

You can download the software packages from the dbsync project library.

- To download the binary installation package of pgsql2pgsql, click here.
- To view instructions on source code compilation of pgsql2pgsql, click here.

Procedure

1. Modify the my.cfg configuration file to configure the connection information of source and destination databases.
   i. Modify the connection information of the source PostgreSQL database.

   > **Note** In the connection information of the source PostgreSQL database, we recommend that you set the user to the owner of the source database.

   ```
   [src.pgsql]
   connect_string = "host=192.168.0.1 dbname=test port=3432  user=test password=pgsql"
   ```

   ii. Modify the connection information of the local temporary PostgreSQL database.

   ```
   [local.pgsql]
   connect_string = "host=192.168.0.2 dbname=test port=3432  user=test2 password=pgsql"
   ```

   iii. Modify the connection information of the destination PostgreSQL database.

   > **Note** You must have the write permissions on the destination table.

   ```
   [desc.pgsql]
   connect_string = "host=192.168.0.2 dbname=test port=3432  user=test3 password=pgsql"
   ```

> **Note**
>
> - If you need to synchronize incremental data, you must have the permissions to create replication slots in the source database.
> - PostgreSQL versions 9.4 and later support logic flow replication, meaning that source databases of the versions support incremental migration. The kernel supports logic flow replication only if you configure the following kernel parameters:
>
>   wal_level = logical
>
>   max_wal_senders = 6
>
>   max_replication_slots = 6

2. Use pgsql2pgsql to perform full database migration.

   ```
   ./pgsql2pgsql
   ```

   By default, the migration program migrates the table data of all users from the source PostgreSQL database to the destination PostgreSQL database.

3. View the status information.

   You can view the status information in a single migration process by connecting to the local temporary database. The information is stored in the db_sync_status table, including the start and end time of the full migration, the start time of the incremental migration, and the status of incremental synchronization.

14.1.3.5.4. Import data by using the \COPY statement

You can use the `\COPY` statement to import the data of local text files into AnalyticDB for PostgreSQL databases. The local text files must be formatted, such as files that use commas (,), semicolons (;), or special characters as delimiters.

Context

- Parallel writing of large amounts of data is not available because the `\COPY` statement writes data in serial using the coordinator node. If you need to import a large amount of data in parallel, you can use the OSS-based data import method.
- The `\COPY` statement is a psql instruction. If you use the database statement `COPY` instead of the `\COPY` statement, you must note that only stdin is supported. This COPY statement does not support file because the root user does not have the superuser permissions to perform operations on files.
- AnalyticDB for PostgreSQL also allows you to use JDBC to execute the COPY statement. The CopyIn method is encapsulated within JDBC. For more information, see Interface CopyIn.
- For more information about how to use the COPY statement, see COPY.

Procedure

1. Import data by using the following sample code:

```
\COPY table [(column [, ...])] FROM {'file' | STDIN}
     [ [WITH]
       [OIDS]
       [HEADER]
       [DELIMITER [ AS ] 'delimiter']
       [NULL [ AS ] 'null string']
       [ESCAPE [ AS ] 'escape' | 'OFF']
       [NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
       [CSV [QUOTE [ AS ] 'quote']
            [FORCE NOT NULL column [, ...]]
       [FILL MISSING FIELDS]
       [[LOG ERRORS [INTO error_table] [KEEP]
        SEGMENT REJECT LIMIT count [ROWS | PERCENT] ]
\COPY {table [(column [, ...])] | (query)} TO {'file' | STDOUT}
      [ [WITH]
        [OIDS]
        [HEADER]
        [DELIMITER [ AS ] 'delimiter']
        [NULL [ AS ] 'null string']
        [ESCAPE [ AS ] 'escape' | 'OFF']
        [CSV [QUOTE [ AS ] 'quote']
             [FORCE QUOTE column [, ...]] ]
      [IGNORE EXTERNAL PARTITIONS ]
```

14.1.4. Databases

14.1.4.1. Overview

The operations based on the Greenplum Database in AnalyticDB for PostgreSQL are the same as those in the Greenplum Database, including schema, supported data types, and user permissions. Except for certain operations exclusive to the Greenplum Database (such as the partition keys and AO tables), you can refer to PostgreSQL for other operations.

References

- Pivotal Greenplum Official Documentation
- Greenplum 4.3 Best Practices

14.1.4.2. Create a database

After you log on to the AnalyticDB for PostgreSQL instance, you can execute SQL statements to create databases.

Similar to PostgreSQL, in AnalyticDB for PostgreSQL you can execute SQL statements to create databases. For example, after **psql** is connected to Greenplum, execute the following statements:

```
=> create database mygpdb;
CREATE DATABASE
=> \c mygpdb
psql (9.4.4, server 8.3devel)
You are now connected to database "mygpdb" as user "mygpdb".
```

14.1.4.3. Create a partition key

AnalyticDB for PostgreSQL is a distributed database and data is distributed across all the data nodes. You must create partition keys to distribute the data. The partition keys are vital to query performance. Partition keys are used to ensure **even data distribution**. Proper selection of keys can significantly improve query performance.

Specify a partition key

In AnalyticDB for PostgreSQL, tables can be distributed across all compute nodes in either hash or random mode. You must specify the partition key when creating a table. Imported data will be distributed to the specific compute node based on the hash value calculated by the partition key.

```
=> create table vtbl(id serial, key integer, value text, shape cuboid, location geometry, comment tex
t) distributed by (key);
CREATE TABLE
```

If you do not specify the partition key (that means a statement without the `distributed by (key)` field), AnalyticDB for PostgreSQL will randomly allocate the ID field by using the round-robin algorithm.

Rules for selecting the partition key

- Select evenly distributed columns or multiple columns to prevent data skew.
- Select fields commonly used for connection operations, especially for highly concurrent statements.
- Select the condition columns that feature high concurrency queries and high filterability.
- Do not use random distribution.

14.1.4.4. Construct data

In some test scenarios, you must construct data to fill the database.

1. Create a function that generates random strings.

```
CREATE OR REPLACE FUNCTION random_string(integer) RETURNS text AS $body$
SELECT array_to_string(array
                     (SELECT substring('0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst
uvwxyz'
                                       FROM (ceil(random()*62))::int
                                       FOR 1)
                      FROM generate_series(1, $1)), '');
$body$
LANGUAGE SQL VOLATILE;
```

2. Create a partition key.

```
CREATE TABLE tbl(id serial, KEY integer, locate geometry, COMMENT text) distributed by (key);
```

3. Construct data.

```
INSERT INTO tbl(KEY, COMMENT, locate)
 SELECT
     KEY,
     COMMENT,
     ST_GeomFromText(locate) AS locate
 FROM
     (SELECT
         (a + 1) AS KEY,
         random_string(ceil(random() * 24)::integer) AS COMMENT,
         'POINT(' || ceil(random() * 36 + 99) || ' ' || ceil(random() * 24 + 50) || ')' AS locate

     FROM
         generate_series(0, 99999) AS a)
     AS t;
```

14.1.4.5. Query data

This topic describes the query statements and how to view the query plans.

Query statement sample

```
=> select * from tbl where key = 751;
| id  | key | value | shape        | locate                                          | comment        |
+-----+-----+-------+--------------+-------------------------------------------------+----------------+
| 751 | 751 | red   | 0101000000000000000C05B4000000000000004A40 | B9hPhjeNWPqV |
(1 row)
Time: 513.101 ms
```

View a query plan

```
=> explain select * from tbl where key = 751;
Gather Motion 1:1  (slice1; segments: 1)  (cost=0.00..1519.28 rows=1 width=53)
  ->  Seq Scan on tbl  (cost=0.00..1519.28 rows=1 width=53)
        Filter: key = 751
Settings:  effective_cache_size=8GB; gp_statistics_use_fkeys=on
Optimizer status: legacy query optimizer
```

14.1.4.6. Manage extensions

You can use extensions to expand database features. AnalyticDB for PostgreSQL enables you to manage extensions.

Extension types

AnalyticDB for PostgreSQL supports the following extensions:

- PostGIS: supports geographic information data.
- MADlib: supports the machine learning function library.
- fuzzystrmatch: supports the fuzzy matching of strings.
- orafunc: compatible with some Oracle functions.
- oss_ext: supports reading data from OSS.
- hll: collects statistics by using the HyperLogLog algorithm.
- pljava: supports compiling user-defined functions (UDF) in the PL/Java language.

- pgcrypto: supports cryptographic hash functions.
- intarray: supports integer array-related functions, operators, and indexes.

Create an extension

Execute the following statements to create an extension:

```
CREATE EXTENSION <extension name>;
CREATE SCHEMA <schema name>;
CREATE EXTENSION IF NOT EXISTS <extension name> WITH SCHEMA <schema name>;
```

> **Note**
>
> Before you create the MADlib extension, you must create the plpythonu extension first.
>
> ```
> CREATE EXTENSION plpythonu;
> CREATE EXTENSION madlib;
> ```

Delete an extension

Execute the following statements to delete an extension:

```
DROP EXTENSION <extension name>;
DROP EXTENSION IF EXISTS <extension name> CASCADE;
```

> **Note** If there are objects dependent on the extension, you must add the CASCADE keyword to delete all dependent objects.

14.1.4.7. Manage users and permissions

This topic describes how to manage users and permissions in AnalyticDB for PostgreSQL.

Manage users

The system prompts you to specify an initial username and password when you create an instance. This initial user is the root user. After the instance is created, you can use the root user account to connect to the database. The system also creates superusers such as aurora and replicator for internal management.

You can run the `\du+` command to view the information of all the users after you connect to the database by using the client tool of PostgreSQL or Greenplum. Example:

```
postgres=> \du+
                                   List of roles
  Role name   |            Attributes             | Member of |  Description
--------------+-----------------------------------+-----------+---------------
 root_user    |                                   |      | rds_superuser
 ...
```

AnalyticDB for PostgreSQL does not provide superuser permissions, but offers a similar role, RDS_SUPERUSER, which is consistent with the permission system of ApsaraDB RDS for PostgreSQL. The root user (such as root_user in the preceding example) has the permissions of the RDS_SUPERUSER role. You can only identify this permission attribute by viewing the user description.

The root user has the following permissions:

- Can create databases and users and perform actions such as LOGIN, excluding the SUPERUSER permissions.
- Can view and modify the data tables of other users and perform actions such as SELECT, UPDATE, DELETE, and changing owners.
- Can view the connection information of other users, cancel their SQL statements, and kill their connections.
- Can create and delete extensions.
- Can create other users with RDS_SUPERUSER permissions. Example:

```
CREATE ROLE root_user2 RDS_SUPERUSER LOGIN PASSWORD 'xyz' ;
```

Manage permissions

You can manage permissions at the database, schema, and table levels. For example, if you want to grant read permissions on a table to a user and revoke their write permissions, you can execute the following statements:

```
GRANT SELECT ON TABLE t1 TO normal_user1;
REVOKE UPDATE ON TABLE t1 FROM normal_user1;
REVOKE DELETE ON TABLE t1 FROM normal_user1;
```

14.1.4.8. Manage JSON data

JavaScript Object Notation (JSON) has become a basic data type in the Internet and IoT fields. For more information about JSON, visit JSON official website. PostgreSQL support for JSON has been well developed. Optimized by Alibaba Cloud, AnalyticDB for PostgreSQL supports the JSON type based on the PostgreSQL syntax.

Check whether the current version supports JSON

Execute the following statement to check whether the current version supports JSON:

```
=> SELECT '""'::json;
```

If the following output is displayed, it indicates the JSON type is supported and the instance is ready for use. If the operation fails, restart the instance.

```
 json
------
 ""
(1 row)
```

If the following output is displayed, it indicates the JSON type is not supported.

```
ERROR:  type "json" does not exist
LINE 1: SELECT '""'::json;
               ^
```

The preceding command converts data from the string type to the JSON type. PostgreSQL supports operations on JSON data based on this conversion.

JSON conversion in the database

Database operations include reading and writing. The written data is typically converted from the string type to the JSON type. The contents of a string must meet the JSON standard, such as strings, digits, arrays, and objects. Example:

String

```
=> SELECT '"hijson"'::json;
 json
-------
 "hijson"
(1 row)
```

`::` is used for explicit type conversion in PostgreSQL, Greenplum, and AnalyticDB for PostgreSQL. The database calls the input function in JSON type during the conversion. Therefore, the JSON format check is performed as follows:

```
=> SELECT '{hijson:1024}'::json;
ERROR:  invalid input syntax for type json
LINE 1: SELECT '{hijson:1024}'::json;
               ^
DETAIL:  Token "hijson" is invalid.
CONTEXT:  JSON data, line 1: {hijson...
=>
```

In the preceding example, `hijson` must be enclosed in double quotation marks (`"`) because JSON requires the KEY value to be a string. A syntax error is returned when `{hijson:1024}` is entered.

Apart from explicit type conversion, database records can also be converted to JSON.

Typically, JSON is not used for a string or a digit, but an object that contains one or more key-value pairs. AnalyticDB for PostgreSQL can support most JSON scenarios after data is converted from the string type to objects. Example:

```
=> select row_to_json(row('{"a":"a"}', 'b'));
           row_to_json
---------------------------------
 {"f1":"{\"a\":\"a\"}","f2":"b"}
(1 row)
=> select row_to_json(row('{"a":"a"}'::json, 'b'));
        row_to_json
---------------------------
 {"f1":{"a":"a"},"f2":"b"}
(1 row)
```

You can see the differences between the string and JSON here. The whole record is conveniently converted into the JSON type.

JSON data types

- Object

  The object is the most frequently used data type in JSON. Example:

  ```
  => select '{"key":"value"}'::json;
          json
    -----------------
     {"key":"value"}
    (1 row)
  ```

- Integer and floating point number

  JSON only supports three data types for numeric values: integer, floating point number, and constant expression. AnalyticDB for PostgreSQL supports all three types.

```
=> SELECT '1024'::json;
 json
------
 1024
(1 row)
=> SELECT '0.1'::json;
 json
------
 0.1
(1 row)
```

The following information is required in some special situations:

```
=> SELECT '1e100'::json;
 json
-------
 1e100
(1 row)
=> SELECT '{"f":1e100}'::json;
    json
-------------
 {"f":1e100}
(1 row)
```

Extra-long numbers are also supported. Example:

```
=> SELECT '9223372036854775808'::json;
        json
---------------------
 9223372036854775808
(1 row)
```

- Array

```
=> SELECT '[[1,2], [3,4,5]]'::json;
        json
------------------
 [[1,2], [3,4,5]]
(1 row)
```

Operators

Operators supported by JSON

```
=> select oprname,oprcode from pg_operator where oprleft = 3114;
 oprname |          oprcode
---------+---------------------------
 ->      | json_object_field
 ->>     | json_object_field_text
 ->      | json_array_element
 ->>     | json_array_element_text
 #>      | json_extract_path_op
 #>>     | json_extract_path_text_op
(6 rows)
```

Basic usage

```
=> SELECT '{"f":"1e100"}'::json -> 'f';
 ? column?
----------
 "1e100"
(1 row)
=> SELECT '{"f":"1e100"}'::json ->> 'f';
 ? column?
----------
 1e100
(1 row)
=> select '{"f2":{"f3":1},"f4":{"f5":99,"f6":"stringy"}}'::json#>array['f4','f6'];
 ? column?
-----------
 "stringy"
(1 row)
=> select '{"f2":{"f3":1},"f4":{"f5":99,"f6":"stringy"}}'::json#>'{f4,f6}';
 ? column?
-----------
 "stringy"
(1 row)
=> select '{"f2":["f3",1],"f4":{"f5":99,"f6":"stringy"}}'::json#>>'{f2,0}';
 ? column?
----------
 f3
(1 row)
```

JSON functions

Supported JSON functions

```
postgres=# \df *json*
                                                  List of functions
   Schema   |           Name            | Result data type |                     Argument data types
|  Type
------------+---------------------------+------------------+-----------------------------------------
------------------+--------
 pg_catalog | array_to_json             | json             | anyarray
| normal
 pg_catalog | array_to_json             | json             | anyarray, boolean
| normal
 pg_catalog | json_array_element        | json             | from_json json, element_index integer
| normal
 pg_catalog | json_array_element_text   | text             | from_json json, element_index integer
| normal
 pg_catalog | json_array_elements       | SETOF json       | from_json json, OUT value json
| normal
 pg_catalog | json_array_length         | integer          | json
| normal
 pg_catalog | json_each                 | SETOF record     | from_json json, OUT key text, OUT value
json              | normal
 pg_catalog | json_each_text            | SETOF record     | from_json json, OUT key text, OUT value
text              | normal
 pg_catalog | json_extract_path         | json             | from_json json, VARIADIC path_elems text
[]                | normal
 pg_catalog | json_extract_path_op      | json             | from_json json, path_elems text[]
| normal
 pg_catalog | json_extract_path_text    | text             | from_json json, VARIADIC path_elems text
[]                | normal
 pg_catalog | json_extract_path_text_op | text             | from_json json, path_elems text[]
| normal
 pg_catalog | json_in                   | json             | cstring
| normal
 pg_catalog | json_object_field         | json             | from_json json, field_name text
| normal
 pg_catalog | json_object_field_text    | text             | from_json json, field_name text
| normal
 pg_catalog | json_object_keys          | SETOF text       | json
| normal
 pg_catalog | json_out                  | cstring          | json
| normal
 pg_catalog | json_populate_record      | anyelement       | base anyelement, from_json json, use_jso
n_as_text boolean | normal
 pg_catalog | json_populate_recordset   | SETOF anyelement | base anyelement, from_json json, use_jso
n_as_text boolean | normal
 pg_catalog | json_recv                 | json             | internal
| normal
 pg_catalog | json_send                 | bytea            | json
| normal
 pg_catalog | row_to_json               | json             | record
| normal
 pg_catalog | row_to_json               | json             | record, boolean
| normal
 pg_catalog | to_json                   | json             | anyelement
| normal
(24 rows)
```

Basic usage

```
=> SELECT array_to_json('{{1,5},{99,100}}'::int[]);
  array_to_json
-----------------
 [[1,5],[99,100]]
(1 row)
=> SELECT row_to_json(row(1,'foo'));
     row_to_json
---------------------
 {"f1":1,"f2":"foo"}
(1 row)
=> SELECT json_array_length('[1,2,3,{"f1":1,"f2":[5,6]},4]');
 json_array_length
-------------------
                 5
(1 row)
=> select * from json_each('{"f1":[1,2,3],"f2":{"f3":1},"f4":null,"f5":99,"f6":"stringy"}') q;
 key |   value
-----+-----------
 f1  | [1,2,3]
 f2  | {"f3":1}
 f4  | null
 f5  | 99
 f6  | "stringy"
(5 rows)
=> select json_each_text('{"f1":[1,2,3],"f2":{"f3":1},"f4":null,"f5":"null"}');
  json_each_text
-------------------
 (f1,"[1,2,3]")
 (f2,"{""f3"":1}")
 (f4,)
 (f5,null)
(4 rows)
=> select json_array_elements('[1,true,[1,[2,3]],null,{"f1":1,"f2":[7,8,9]},false]');
  json_array_elements
-----------------------
 1
 true
 [1,[2,3]]
 null
 {"f1":1,"f2":[7,8,9]}
 false
(6 rows)
create type jpop as (a text, b int, c timestamp);
=> select * from json_populate_record(null::jpop,'{"a":"blurfl","x":43.2}', false) q;
   a    | b | c
--------+---+---
 blurfl |   |
(1 row)
=>     select * from json_populate_recordset(null::jpop,'[{"a":"blurfl","x":43.2},{"b":3,"c":"2012-01
-20 10:42:53"}]',false) q;
   a    | b |            c
--------+---+--------------------------
 blurfl |   |
        | 3 | Fri Jan 20 10:42:53 2012
(2 rows)
```

Code examples

Create a table

```
create table tj(id serial, ary int[], obj json, num integer);
=> insert into tj(ary, obj, num) values('{1,5}'::int[], '{"obj":1}', 5);
INSERT 0 1
=> select row_to_json(q) from (select id, ary, obj, num from tj) as q;
               row_to_json
-------------------------------------------
 {"f1":1,"f2":[1,5],"f3":{"obj":1},"f4":5}
(1 row)
=> insert into tj(ary, obj, num) values('{2,5}'::int[], '{"obj":2}', 5);
INSERT 0 1
=> select row_to_json(q) from (select id, ary, obj, num from tj) as q;
               row_to_json
-------------------------------------------
 {"f1":1,"f2":[1,5],"f3":{"obj":1},"f4":5}
 {"f1":2,"f2":[2,5],"f3":{"obj":2},"f4":5}
(2 rows)
```

Join multiple tables

```
create table tj2(id serial, ary int[], obj json, num integer);
=> insert into tj2(ary, obj, num) values('{2,5}'::int[], '{"obj":2}', 5);
INSERT 0 1
=> select * from tj, tj2 where tj.obj->>'obj' = tj2.obj->>'obj';
 id |  ary  |    obj    | num | id |  ary  |    obj    | num
----+-------+-----------+-----+----+-------+-----------+-----
  2 | {2,5} | {"obj":2} |   5 |  1 | {2,5} | {"obj":2} |   5
(1 row)
=> select * from tj, tj2 where json_object_field_text(tj.obj, 'obj') = json_object_field_text(tj2.obj
, 'obj');
 id |  ary  |    obj    | num | id |  ary  |    obj    | num
----+-------+-----------+-----+----+-------+-----------+-----
  2 | {2,5} | {"obj":2} |   5 |  1 | {2,5} | {"obj":2} |   5
(1 row)
```

Use the JSON function index

```
CREATE TEMP TABLE test_json (
       json_type text,
       obj json
);
=> insert into test_json values('aa', '{"f2":{"f3":1},"f4":{"f5":99,"f6":"foo"}}');
INSERT 0 1
=> insert into test_json values('cc', '{"f7":{"f3":1},"f8":{"f5":99,"f6":"foo"}}');
INSERT 0 1
=> select obj->'f2' from test_json where json_type = 'aa';
 ? column?
----------
 {"f3":1}
(1 row)
=> create index i on test_json (json_extract_path_text(obj, '{f4}'));
CREATE INDEX
=> select * from test_json where json_extract_path_text(obj, '{f4}') = '{"f5":99,"f6":"foo"}';
 json_type |                 obj
-----------+---------------------------------------
 aa        | {"f2":{"f3":1},"f4":{"f5":99,"f6":"foo"}}
(1 row)
```

> **Note**
>
> JSON data cannot be used as the partition key and does not support JSON aggregate functions.

Example of using Python to access the database:

```
#! /bin/env python
import time
import json
import psycopg2
def gpquery(sql):
    conn = None
    try:
        conn = psycopg2.connect("dbname=sanity1x2")
        conn.autocommit = True
        cur = conn.cursor()
        cur.execute(sql)
        return cur.fetchall()
    except Exception as e:
        if conn:
            try:
                conn.close()
            except:
                pass
            time.sleep(10)
        print e
    return None
def main(_):
    sql = "select obj from tj;"
    #rows = Connection(host, port, user, pwd, dbname).query(sql)
    rows = gpquery(sql)
    for row in rows:
        print json.loads(row[0])
if __name__ == '__main__':
    main()
```

14.1.4.9. Use HyperLogLog

AnalyticDB for PostgreSQL is highly optimized by Alibaba Cloud, and not only has the features of Greenplum Database, but also supports HyperLogLog. It is suitable for industries such as Internet advertising and estimation analysis that require quick estimation of business metrics such as PV and UV.

Create a HyperLogLog extension

You can execute the following statement to create a HyperLogLog extension:

```
CREATE EXTENSION hll;
```

Basic types

- Execute the following statement to create a table containing the hll field:

  ```
  create table agg (id int primary key,userids hll);
  ```

- Execute the following statement to convert int to hll_hashval:

  ```
  select 1::hll_hashval;
  ```

Basic operators

- The hll type supports =, !=, <>, ||, and #.

```
select hll_add_agg(1::hll_hashval) = hll_add_agg(2::hll_hashval);
select hll_add_agg(1::hll_hashval) || hll_add_agg(2::hll_hashval);
select #hll_add_agg(1::hll_hashval);
```

- The hll_hashval type supports =, !=, and <>.

```
select 1::hll_hashval = 2::hll_hashval;
select 1::hll_hashval <> 2::hll_hashval;
```

Basic functions

- Hash functions such as Hll_hash_boolean, hll_hash_smallint, and hll_hash_bigint.

```
select hll_hash_boolean(true);
select hll_hash_integer(1);
```

- hll_add_agg: converts the int format to the hll format.

```
select hll_add_agg(1::hll_hashval);
```

- hll_union: aggregates the hll fields.

```
select hll_union(hll_add_agg(1::hll_hashval),hll_add_agg(2::hll_hashval));
```

- hll_set_defaults: sets the precision.

```
select hll_set_defaults(15,5,-1,1);
```

- hll_print: displays debug information.

```
select hll_print(hll_add_agg(1::hll_hashval));
```

Examples

```
create table access_date (acc_date date unique, userids hll);
insert into access_date select current_date, hll_add_agg(hll_hash_integer(user_id)) from generate_ser
ies(1,10000) t(user_id);
insert into access_date select current_date-1, hll_add_agg(hll_hash_integer(user_id)) from generate_s
eries(5000,20000) t(user_id);
insert into access_date select current_date-2, hll_add_agg(hll_hash_integer(user_id)) from generate_s
eries(9000,40000) t(user_id);
postgres=# select #userids from access_date where acc_date=current_date;
     ? column?
------------------
 9725.85273370708
(1 row)
postgres=# select #userids from access_date where acc_date=current_date-1;
     ? column?
------------------
 14968.6596883279
(1 row)
postgres=# select #userids from access_date where acc_date=current_date-2;
     ? column?
------------------
 29361.5209149911
(1 row)
```

14.1.4.10. Use the CREATE LIBRARY statement

AnalyticDB for PostgreSQL introduces the CREATE LIBRARY and DROP LIBRARY statements to allow you to import custom software packages.

Syntax

```
CREATE LIBRARY library_name LANGUAGE [JAVA] FROM oss_location OWNER ownername
CREATE LIBRARY library_name LANGUAGE [JAVA] VALUES file_content_hex OWNER ownername
DROP LIBRARY library_name
```

Parameters

Parameter	Description
library_name	The name of the library to be installed. If the library to be installed has the same name as an existing library, you must delete the existing library before installing the new one.
LANGUAGE [JAVA]	The programming language to be used. Only PL/Java is supported.
oss_location	The location of the package. You can specify the OSS bucket and object names. Only one object can be specified and the specified object cannot be a compressed file. The format is as follows: `oss://oss_endpoint filepath=[folder/[folder/]...]/file_name id=userossid key=userosskey bucket=ossbucket`
file_content_hex	The content of the file. The byte stream is in hexadecimal notation. For example, 73656c6563742031 indicates the hexadecimal byte stream of "select 1". You can use this syntax to import packages without using OSS.
ownername	Specifies the user.
DROP LIBRARY	Deletes a library.

Examples

- Example 1: Install a JAR package named analytics.jar.

```
create library example language java from 'oss://oss-cn-hangzhou.aliyuncs.com filepath=analytics.jar id=xxx key=yyy bucket=zzz';
```

- Example 2: Import the file content with the byte stream in hexadecimal notation.

```
create library pglib LANGUAGE java VALUES '73656c6563742031' OWNER "myuser";
```

- Example 3: Delete a library.

```
drop library example;
```

- Example 4: View installed libraries.

```
select name, lanname from pg_library;
```

14.1.4.11. Create and use the PL/Java UDF

AnalyticDB for PostgreSQL allows you to compile and upload JAR software packages written in PL/Java language, and use these JAR packages to create user-defined functions (UDFs). The PL/Java language supported by AnalyticDB for PostgreSQL is Community Edition PL/Java 1.5.0 and the JVM version is 1.8. This topic describes how to create a PL/Java UDF. For more information about PL/Java examples, see PL/Java code. For more information about the compiling method, see PL/Java documentation.

Procedure

1. In AnalyticDB for PostgreSQL, execute the following statement to create a PL/Java extension. You only need to execute the statement once for each database.

```
create extension pljava;
```

2. Compile the UDF based on your business needs. For example, you can use the following code to compile the Test.java file:

```
public class Test
 {
     public static String substring(String text, int beginIndex,
                int endIndex)
             {
                     try {
                         Process process = null;
                         process = Runtime.getRuntime().exec("ech Test running");
                     } catch (Exception e) {
                        return "" + e;
                     }
                     return text.substring(beginIndex, endIndex);
             }
 }
```

3. Compile the manifest.txt file:

```
 Manifest-Version: 1.0
 Main-Class: Test
 Specification-Title: "Test"
 Specification-Version: "1.0"
 Created-By: 1.7.0_99
 Build-Date: 01/20/2016 21:00 AM
```

4. Run the following commands to compile and package the program:

```
javac Test.java
 jar cfm analytics.jar manifest.txt Test.class
```

5. Upload the analytics.jar file generated in step 4 to OSS by using the following OSS console command.

```
osscmd put analytics.jar oss://zzz
```

6. In AnalyticDB for PostgreSQL, execute the CREATE LIBRARY statement to import the file to AnalyticDB for PostgreSQL:

```
create library example language java from 'oss://oss-cn-hangzhou.aliyuncs.com filepath=analytics.
jar id=xxx key=yyy bucket=zzz';
```

> **Note** You can only use the filepath variable in the CREATE LIBRARY statement to import files one at a time. Additionally, the CREATE LIBRARY statement also supports byte streams to import files without using OSS. For more information, see Use the CREATE LIBRARY statement.

7. In AnalyticDB for PostgreSQL, execute the following statements to create and use the UDF.

```
create table temp (a varchar) distributed randomly;
 insert into temp values ('my string');
 create or replace function java_substring(varchar, int, int) returns varchar as 'Test.substring'
language java;
 select java_substring(a, 1, 5) from temp;
```

14.1.5. Table

14.1.5.1. Create a table

You can create tables within your databases.

Syntax

The complete syntax for creating a table is as follows. Depending on your business needs, not all clauses will be required. Use the clauses that can fulfill your business needs.

```
CREATE [[GLOBAL | LOCAL] {TEMPORARY | TEMP}] TABLE table_name (
[ { column_namedata_type [ DEFAULT default_expr ]
   [column_constraint [ ... ]
[ ENCODING ( storage_directive [,...] ) ]
]
   | table_constraint
   | LIKE other_table [{INCLUDING | EXCLUDING}
                      {DEFAULTS | CONSTRAINTS}] ...}
   [, ... ] ]
   )
   [ INHERITS ( parent_table [, ... ] ) ]
   [ WITH ( storage_parameter=value [, ... ] )
   [ ON COMMIT {PRESERVE ROWS | DELETE ROWS | DROP} ]
   [ TABLESPACE tablespace ]
   [ DISTRIBUTED BY (column, [ ... ] ) | DISTRIBUTED RANDOMLY ]
   [ PARTITION BY partition_type (column)
       [ SUBPARTITION BY partition_type (column) ]
          [ SUBPARTITION TEMPLATE ( template_spec ) ]
       [...]
    ( partition_spec )
        | [ SUBPARTITION BY partition_type (column) ]
          [...]
    ( partition_spec
      [ ( subpartition_spec
           [(...)]
        ) ]
    )
```

The column_constraint clause can be defined as follows:

```
[CONSTRAINT constraint_name]
   NOT NULL | NULL
   | UNIQUE [USING INDEX TABLESPACE tablespace]
            [WITH ( FILLFACTOR = value )]
   | PRIMARY KEY [USING INDEX TABLESPACE tablespace]
                 [WITH ( FILLFACTOR = value )]
   | CHECK ( expression )
   | REFERENCES table_name [ ( column_name [, ... ] ) ]
            [ key_match_type ]
            [ key_action ]
```

The storage_directive clause of columns can be defined as follows:

```
COMPRESSTYPE={ZLIB | QUICKLZ | RLE_TYPE | NONE}
 [COMPRESSLEVEL={0-9} ]
 [BLOCKSIZE={8192-2097152} ]
```

The storage_parameter clause of tables can be defined as follows:

```
APPENDONLY={TRUE|FALSE}
BLOCKSIZE={8192-2097152}
ORIENTATION={COLUMN|ROW}
CHECKSUM={TRUE|FALSE}
COMPRESSTYPE={ZLIB|QUICKLZ|RLE_TYPE|NONE}
COMPRESSLEVEL={0-9}
FILLFACTOR={10-100}
OIDS[=TRUE|FALSE]
```

The table_constraint clause can be defined as follows:

```
[CONSTRAINT constraint_name]
UNIQUE ( column_name [, ... ] )
       [USING INDEX TABLESPACE tablespace]
       [WITH ( FILLFACTOR=value )]
| PRIMARY KEY ( column_name [, ... ] )
              [USING INDEX TABLESPACE tablespace]
              [WITH ( FILLFACTOR=value )]
| CHECK ( expression )
| FOREIGN KEY ( column_name [, ... ] )
         REFERENCES table_name [ ( column_name [, ... ] ) ]
         [ key_match_type ]
         [ key_action ]
         [ key_checking_mode ]
```

Valid values of key_match_type:

```
  MATCH FULL
| SIMPLE
```

Valid values of key_action:

```
  ON DELETE
| ON UPDATE
| NO ACTION
| RESTRICT
| CASCADE
| SET NULL
| SET DEFAULT
```

Valid values of key_checking_mode:

```
  DEFERRABLE
| NOT DEFERRABLE
| INITIALLY DEFERRED
| INITIALLY IMMEDIATE
```

Valid values of partition_type:

```
    LIST
  | RANGE
```

The partition_specification clause can be defined as follows:

```
partition_element [, ...]
```

The partition_element clause can be defined as follows:

```
   DEFAULT PARTITION name
  | [PARTITION name] VALUES (list_value [,...] )
  | [PARTITION name]
     START ([datatype] 'start_value') [INCLUSIVE | EXCLUSIVE]
     [ END ([datatype] 'end_value') [INCLUSIVE | EXCLUSIVE] ]
     [ EVERY ([datatype] [number | INTERVAL] 'interval_value') ]
  | [PARTITION name]
     END ([datatype] 'end_value') [INCLUSIVE | EXCLUSIVE]
     [ EVERY ([datatype] [number | INTERVAL] 'interval_value') ]
[ WITH ( partition_storage_parameter=value [, ... ] ) ]
[ TABLESPACE tablespace ]
```

The subpartition_spec or template_spec clause can be defined as follows:

```
subpartition_element [, ...]
```

The subpartition_element clause can be defined as follows:

```
   DEFAULT SUBPARTITION name
  | [SUBPARTITION name] VALUES (list_value [,...] )
  | [SUBPARTITION name]
     START ([datatype] 'start_value') [INCLUSIVE | EXCLUSIVE]
     [ END ([datatype] 'end_value') [INCLUSIVE | EXCLUSIVE] ]
     [ EVERY ([datatype] [number | INTERVAL] 'interval_value') ]
  | [SUBPARTITION name]
     END ([datatype] 'end_value') [INCLUSIVE | EXCLUSIVE]
     [ EVERY ([datatype] [number | INTERVAL] 'interval_value') ]
[ WITH ( partition_storage_parameter=value [, ... ] ) ]
[ TABLESPACE tablespace ]
```

The storage_parameter clause of partitions can be defined as follows:

```
   APPENDONLY={TRUE|FALSE}
   BLOCKSIZE={8192-2097152}
   ORIENTATION={COLUMN|ROW}
   CHECKSUM={TRUE|FALSE}
   COMPRESSTYPE={ZLIB|QUICKLZ|RLE_TYPE|NONE}
   COMPRESSLEVEL={1-9}
   FILLFACTOR={10-100}
   OIDS[=TRUE|FALSE]
```

Parameters

The Table creation parameters table describes the key parameters for creating a table.

Table creation parameters

Parameter	Description
TABLE_NAME	The name of the table to be created.
column_name	The name of a column to be created in the new table.
data_type	The data type of the column. For columns that contain textual data, set the data type to VARCHAR or TEXT. We do not recommend the CHAR type.
DEFAULT default_expr	Specifies a default value for the column. The system will assign this default value to all columns that do not have a value. The default values can be any variable-free expression. Subqueries or cross-references to other columns in the table are not allowed. The data type of the default expression must match the data type of the column. If a column does not have a default value, the default value is null.
ENCODING storage_directive	Specifies the type of compression and block size for the column data. This clause is valid only for append-optimized, column-oriented tables. Column compression settings are inherited from the table level to the partition level to the sub-partition level. The lowest-level settings have priority over inherited settings.
INHERITS	Specifies that all columns in the new table automatically inherit a parent table. You can use INHERITS to create a persistent relationship between the new child table and its parent table. Schema modifications to the parent table are applied to the child table as well. When the parent table is also scanned, the data of the child table is scanned as well.
LIKE other_table	Specifies a table from which the new table automatically copies all column names, data types, NOT NULL constraints, and distribution policies. Storage properties such as append-optimized or partition structure are not copied. Unlike INHERITS, the new table is completely decoupled from the original table after the new table is created.
CONSTRAINT constraint_name	Configures a column or table constraint. When a constraint is violated, the constraint name will be displayed in the error message. Constraint names can be used to communicate helpful information to client applications. Constraint names that contain spaces must be enclosed by double quotation marks ('').
WITH (storage_option=value)	Configures storage options for the table or its indexes.
ON COMMIT	The operation that the system performs on the temporary tables at the end of a transaction. Valid values: • **PRESERVE ROWS**: No special action is taken. The data will be retained after the transaction is complete. The data will only be released when the session is disconnected. • **DELETE ROWS**: All rows in the temporary table are deleted. • **DROP**: The temporary table is deleted.
TABLESPACE tablespace	Specifies the name of the tablespace in which the new table is to be created. If not specified, the default tablespace of the database is used.

Parameter	Description
DISTRIBUTED BY	Specifies the distribution policy for the database. • DISTRIBUTED BY (column, [...]): specifies the partition key. The system uses hash distribution based on the distribution key. To evenly distribute data, you must set the partition key to the primary key of the table or a unique column or a set of columns. • DISTRIBUTED RANDOMLY: distributes data randomly. Note We recommend that you do not use random distribution.
PARTITION BY	Configures the partition key to partition the table. Partitioning large tables improves data access efficiency. To partition a table is to create a top-level (parent) table and multiple lower-level (child) tables. A parent table is always empty when the partition table is created. Data is stored in the lowest-level child tables. In a multi-level partition table, data is only stored in the lowest-level sub-partitions. Valid values: RANGE, LIST, and a combination of the two.
SUBPARTITION BY	Configures a multi-level partitioned table.
SUBPARTITION TEMPLATE	You can specify a sub-partition template to create sub-partitions (lower-level child tables). This sub-partition template is applied to all parent partitions to ensure the same sub-partition structure.

Examples

Create a table and configure the partition key. The primary key is the default partition key in AnalyticDB for PostgreSQL.

```
CREATE TABLE films (
code        char(5) CONSTRAINT firstkey PRIMARY KEY,
title       varchar(40) NOT NULL,
did         integer NOT NULL,
date_prod   date,
kind        varchar(10),
len         interval hour to minute
);
CREATE TABLE distributors (
did    integer PRIMARY KEY DEFAULT nextval('serial'),
name   varchar(40) NOT NULL CHECK (name <> '')
);
```

Create a compressed table and configure the partition key.

```
CREATE TABLE sales (txn_id int, qty int, date date)
WITH (appendonly=true, compresslevel=5)
DISTRIBUTED BY (txn_id);
```

Use sub-partition templates of each level and the default partition to create a three-level partition table.

```
CREATE TABLE sales (id int, year int, month int, day int,
region text)
DISTRIBUTED BY (id)
PARTITION BY RANGE (year)
  SUBPARTITION BY RANGE (month)
    SUBPARTITION TEMPLATE (
       START (1) END (13) EVERY (1),
       DEFAULT SUBPARTITION other_months )
  SUBPARTITION BY LIST (region)
    SUBPARTITION TEMPLATE (
       SUBPARTITION usa VALUES ('usa'),
       SUBPARTITION europe VALUES ('europe'),
       SUBPARTITION asia VALUES ('asia'),
       DEFAULT SUBPARTITION other_regions)
( START (2008) END (2016) EVERY (1),
  DEFAULT PARTITION outlying_years);
```

14.1.5.2. Principles and scenarios of row store, column store, heap tables, and AO tables

AnalyticDB for PostgreSQL supports row store, column store, heap tables, and AO tables. This topic describes their principles and scenarios.

Row store and column store

Comparison

Dimension	Row store	Column store
Definition	Row store stores data in the form of rows. Each row is a tuple. To read a column, you must deform all of the columns that precede the target column. Because of this, the costs for accessing the first and the last columns are different.	Column store stores data as columns corresponding to a file or a batch of files. The cost of reading any column is the same. However, if you need to read multiple columns, you must access multiple files. The more columns you access, the higher the overheads are.
Compression ratio	Low.	High.
Cost of reading any column	Columns with larger column numbers cost more.	Same.
Vector computing and JIT architecture	Not suitable. Not suitable for batch computation.	Suitable. More efficient when accessing and obtaining statistics of a batch of data.

Dimension	Row store	Column store
Scenarios	If you need to perform a large number of update and delete operations due to OLTP requirements such as when querying table details where multiple columns are returned, you can use row store. You can use partition tables if you have diversified requirements. For example, if you need to partition the data based on time, you can use row store to query the details of recent data and use column store to obtain more statistics from historical data.	You can use column store if you need data statistics because of the OLAP requirements. If you need a higher compression ratio, you can use column store.

Heap tables

A heap table is heap storage. All changes to the heap table generate redo logs that can be used to restore data by time point. However, heap tables cannot implement logical incremental backup because any data block in the table may be changed and it is not convenient to record the position by using the heap storage.

Commit and redo logs are used to ensure reliability when transactions are finished. You can also implement redundancy by building secondary nodes through redo logs.

Append-optimized (AO) tables

AO tables are used to append data for storage. When you delete the updated data, you can use another bitmap file to mark the row to be deleted and use the bit and offset to determine whether a row is deleted.

When the transaction is finished, you must call the fsync function to record the offset of the data block that performs the last write operation. Even if the data block only contains one record, a new data block will be appended for the next transaction. The data block is synchronized to the secondary node for data redundancy.

AO tables are not suitable for small transactions because the fsync function is called at the end of each transaction, and this data block will not be reused even if there is space left.

AO tables are suitable for OLAP scenarios, batch data writing, high compression ratio, and logical backup that supports incremental backup. During backup, you only need to record the offset from the backup and the bitmap deletion mark for each full backup.

Usage scenarios of heap tables

- When multiple small transactions are handled, use a heap table.
- When you need to restore data by time point, use a heap table.

Usage scenarios of AO tables

- When you need to use column store, use an AO table.
- When data is written in batches, use an AO table.

14.1.5.3. Enable the column store and compression features

If you want to improve performance, speed up data import, or reduce costs for tables with infrequent updates and multiple fields, we recommend that you use column store and compression. This will increase the compression ratio threefold to ensure faster performance and import speed.

To enable the column store and compression features, you must specify the column store and compression options when creating a table. For example, you can add the following clause to the CREATE statement to enable the two features. For more information about the table creation syntax, see Create a table.

```
with (APPENDONLY=true, ORIENTATION=column, COMPRESSTYPE=zlib, COMPRESSLEVEL=5, BLOCKSIZE=1048576, OIDS=false)
```

Note AnalyticDB for PostgreSQL only supports zlib and RLE_TYPE compression algorithms. If you specify the quicklz algorithm, it is automatically converted to zlib.

14.1.5.4. Add a field to a column store table and set the default value

This topic describes how to add a field to a column store table and set the default value for the field, and how to use the ANALYZE statement to view the impact of updated data on the size of the column store table.

Context

In a column store table, each column is stored as a file, and two columns in the same row correspond to each other by using the offset. For example, if you add two fields of the INT8 type, you can quickly locate column B from column A by using the offset.

When you add the field, AO tables are not rewritten. If an AO table contains the records of deleted data, the added field must be filled with the deleted records before using the offset.

Procedure

1. Create three AO column store tables.

```
postgres=# create table tbl1 (id int, info text) with (appendonly=true, blocksize=8192, compresstype=none, orientation=column);
NOTICE:  Table doesn't have 'DISTRIBUTED BY' clause -- Using column named 'id' as the Greenplum Database data distribution key for this table.
HINT:  The 'DISTRIBUTED BY' clause determines the distribution of data. Make sure column(s) chosen are the optimal data distribution key to minimize skew.
CREATE TABLE
postgres=# create table tbl2 (id int, info text) with (appendonly=true, blocksize=8192, compresstype=none, orientation=column);
NOTICE:  Table doesn't have 'DISTRIBUTED BY' clause -- Using column named 'id' as the Greenplum Database data distribution key for this table.
HINT:  The 'DISTRIBUTED BY' clause determines the distribution of data. Make sure column(s) chosen are the optimal data distribution key to minimize skew.
CREATE TABLE
postgres=# create table tbl3 (id int, info text) with (appendonly=true, blocksize=8192, compresstype=none, orientation=column);
NOTICE:  Table doesn't have 'DISTRIBUTED BY' clause -- Using column named 'id' as the Greenplum Database data distribution key for this table.
HINT:  The 'DISTRIBUTED BY' clause determines the distribution of data. Make sure column(s) chosen are the optimal data distribution key to minimize skew.
CREATE TABLE
```

2. Insert 10 million entries to the first two tables and 20 million entries to the third one.

```
postgres=# insert into tbl1 select generate_series(1,10000000),'test';
INSERT 0 10000000
postgres=# insert into tbl2 select generate_series(1,10000000),'test';
INSERT 0 10000000
postgres=# insert into tbl3 select generate_series(1,20000000),'test';
INSERT 0 20000000
```

3. Analyze the tables and display their sizes.

```
postgres=# analyze tbl1;
ANALYZE
postgres=# analyze tbl2;
ANALYZE
postgres=# analyze tbl3;
ANALYZE
postgres=# select pg_size_pretty(pg_relation_size('tbl1'));
 pg_size_pretty
----------------
 88 MB
(1 row)
postgres=# select pg_size_pretty(pg_relation_size('tbl2'));
 pg_size_pretty
----------------
 88 MB
(1 row)
postgres=# select pg_size_pretty(pg_relation_size('tbl3'));
 pg_size_pretty
----------------
 173 MB
(1 row)
```

4. Update all the data in the first table. Display the table size after the update. The size is twice as large as the size before the update.

```
postgres=# update tbl1 set info='test';
UPDATE 10000000
postgres=# analyze tbl1;
ANALYZE
postgres=# select pg_size_pretty(pg_relation_size('tbl1'));
 pg_size_pretty
----------------
 173 MB
(1 row)
```

5. Add fields to the three tables and set the default values.

```
postgres=# alter table tbl1 add column c1 int8 default 1;
ALTER TABLE
postgres=# alter table tbl2 add column c1 int8 default 1;
ALTER TABLE
postgres=# alter table tbl3 add column c1 int8 default 1;
ALTER TABLE
```

6. Analyze the tables and view the table sizes.

```
postgres=# analyze tbl1;
ANALYZE
postgres=# analyze tbl2;
ANALYZE
postgres=# analyze tbl3;
ANALYZE
postgres=# select pg_size_pretty(pg_relation_size('tbl1'));
 pg_size_pretty
----------------
 325 MB
(1 row)
postgres=# select pg_size_pretty(pg_relation_size('tbl2'));
 pg_size_pretty
----------------
 163 MB
(1 row)
postgres=# select pg_size_pretty(pg_relation_size('tbl3'));
 pg_size_pretty
----------------
 325 MB
(1 row)
```

When you add fields to the AO tables, the number of entries in the existing files will prevail. Even if all the entries are deleted, you must initialize the original data in the newly added fields.

14.1.5.5. Configure the table partition

For fact tables or large-sized tables in the database, we recommend that you configure table partitions.

Configure the table partition

You can use the table partitioning feature to delete data by using the `ALTER TABLE DROP PARTITION` statement to delete all the data in a partition, and import data by using the `ALTER TABLE EXCHANGE PARTITION` statement to add a new data partition on a regular basis.

AnalyticDB for PostgreSQL supports range partitioning, list partitioning, and composite partitioning. Range partitioning only supports partitioning by fields of the numeric or datetime data types.

The following example shows a table that uses range partitioning.

```
CREATE TABLE LINEITEM (
L_ORDERKEY              BIGINT NOT NULL,
L_PARTKEY               BIGINT NOT NULL,
L_SUPPKEY               BIGINT NOT NULL,
L_LINENUMBER    INTEGER,
L_QUANTITY              FLOAT8,
L_EXTENDEDPRICE FLOAT8,
L_DISCOUNT              FLOAT8,
L_TAX                   FLOAT8,
L_RETURNFLAG    CHAR(1),
L_LINESTATUS    CHAR(1),
L_SHIPDATE              DATE,
L_COMMITDATE    DATE,
L_RECEIPTDATE   DATE,
L_SHIPINSTRUCT  CHAR(25),
L_SHIPMODE              CHAR(10),
L_COMMENT               VARCHAR(44)
)  WITH  (APPENDONLY=true, ORIENTATION=column, COMPRESSTYPE=zlib, COMPRESSLEVEL=5, BLOCKSIZE=1048576,
OIDS=false) DISTRIBUTED BY (l_orderkey)
PARTITION BY RANGE (L_SHIPDATE) (START (date '1992-01-01') INCLUSIVE END (date '2000-01-01') EXCLUSIV
E EVERY (INTERVAL '1 month' ));
```

Principles of table partitioning

The purpose of partitioning is to minimize the amount of data that needs to be scanned during a query, so partitions must be associated with the query conditions.

- Principle 1: Select the fields related to the query conditions to configure partitions and reduce the amount of data to be scanned.
- Principle 2: When multiple query conditions exist, configure sub-partitions to further reduce the amount of data to be scanned.

14.1.5.6. Configure the sort key

A sort key is an attribute of a table. Data on disks is stored in the order of the sort key.

Context

Sort keys have two major advantages:

- Speed up and optimize column-store operations. The min and max meta information the system collects seldom overlaps with each other, which features good filterability.
- Eliminate the need to perform ORDER BY and GROUP BY operations. The data directly read from the disk is ordered as required by the sorting conditions.

Create a table

```
Command:     CREATE TABLE
Description: define a new table
Syntax:
CREATE [[GLOBAL | LOCAL] {TEMPORARY | TEMP}] TABLE table_name (
[ { column_name data_type [ DEFAULT default_expr ]     [column_constraint [ ... ]
[ ENCODING ( storage_directive [,...] ) ]
]
   | table_constraint
   | LIKE other_table [{INCLUDING | EXCLUDING}
                      {DEFAULTS | CONSTRAINTS}] ...}
   [, ... ] ]
   [column_reference_storage_directive [, ] ]
   )
   [ INHERITS ( parent_table [, ... ] ) ]
   [ WITH ( storage_parameter=value [, ... ] )
   [ ON COMMIT {PRESERVE ROWS | DELETE ROWS | DROP} ]
   [ TABLESPACE tablespace ]
   [ DISTRIBUTED BY (column, [ ... ] ) | DISTRIBUTED RANDOMLY ]
   [ SORTKEY (column, [ ... ] )]
   [ PARTITION BY partition_type (column)
       [ SUBPARTITION BY partition_type (column) ]
          [ SUBPARTITION TEMPLATE ( template_spec ) ]
       [...]
    ( partition_spec )
        | [ SUBPARTITION BY partition_type (column) ]
          [...]
    ( partition_spec
      [ ( subpartition_spec
           [(...)]
         ) ]
)
```

Examples:

```
create table test(date text, time text, open float, high float, low float, volume int) with(APPENDONLY=true,ORIENTATION=column) sortkey (volume);
```

Sort the table

```
VACUUM SORT ONLY [tablename]
```

Modify the sort key

This statement only modifies the catalog and does not sort data. You must execute the `VACUUM SORT ONLY` statement to sort the table.

```
ALTER [[GLOBAL | LOCAL] {TEMPORARY | TEMP}] TABLE table_name SET SORTKEY (column, [ ... ] )
```

Examples:

```
alter table test set sortkey (high,low);
```

14.1.6. Best practices

14.1.6.1. Configure memory and load parameters

You must configure memory and load parameters to improve database stability.

Background information

AnalyticDB for PostgreSQL is an MPP database with high computational and resource requirements. It consumes all of the resources provided to it, allowing AnalyticDB for PostgreSQL to have higher processing speeds but making it very easy to reach its limits.

The worst-case scenario in the event of the CPU, network, or hard disk exceeding its limits is a hardware bottleneck. However, in the event that memory is completely consumed, the database may crash.

How to avoid OOM errors

Out of memory (OOM) indicates that the system is unable to provide sufficient memory requested by a process. The following prompt appears when OOM errors occur:

```
Out of memory (seg27 host.example.com pid=47093) VM Protect failed to allocate 4096 bytes, 0 MB avail
able
```

Causes

Possible causes of the OOM error include:

- The memory of the database node is insufficient.
- Kernel parameters related to the memory of the operating system are incorrectly configured.
- Data skew has occurred, causing a compute node to request a large amount of memory.
- Query skew has occurred. For example, if the grouping fields of some aggregate or window functions are not distribution keys, the data must be redistributed. After redistribution, data will be skewed in a certain computer node and result in the node requesting a large amount of memory.

Solutions

1. Modify the queries to request less memory.
2. Use the resource queue provided by AnalyticDB for PostgreSQL to limit the number of concurrent queries. Reduce the number of queries executed within the cluster at the same time to reduce the overall memory requested by the system.
3. Reduce the number of compute nodes deployed on a host. For example, deploy 8 compute nodes instead of 16 compute nodes on a host with 128 GB of memory. This allows each compute node to use twice the amount of memory compared with the latter.
4. Increase the memory of a host.
5. Set the gp_vmem_protect_limit parameter to limit the maximum VMEM that can be used by a single compute node. The memory size of a single host and the number of compute nodes deployed on the host determine the maximum memory size that a single compute node can use on average.
6. For SQL statements that have unpredictable memory usage, you can set the statement_mem parameter in the session to limit the memory usage of a single SQL statement, so as to prevent a single SQL statement from consuming all available memory.
7. Set the statement_mem parameter at the database level to apply to all the sessions in the database.
8. Use the resource queue to limit the maximum memory usage of the resource group. Add database users to the resource group to limit the overall memory used by these users.

Configure memory-related parameters

Properly configuring the operating system, database parameters, and resource queue can effectively reduce the probability of OOM.

When calculating the average memory usage of a single compute node on a single host, you must consider both the primary and secondary compute nodes. When the cluster encounters a host failure, the system will switch the service from primary compute nodes to the corresponding secondary compute nodes. During this time, the number of compute nodes on the host will be greater than usual. Therefore, you must consider the number of resources that will be occupied by the secondary compute nodes during failover.

The following tables describe how to configure parameters of the operating system kernel and database to avoid OOM.

The following Operating system kernel parameters table describes the parameter configuration of the operating system kernel.

Operating system kernel parameters

Parameter	Description
huge page	Do not configure the huge page parameter of the system. AnalyticDB for PostgreSQL does not support the latest version of PostgreSQL and therefore does not support the huge page feature. The huge page parameter locks a part of the allocated memory. Database nodes will not be able to use this part of the memory.
vm.overcommit_memory	If you use the swap space, set this parameter to 2. If you do not use the swap space, set this parameter to 0. Valid values: • 0: The requested memory space cannot exceed the difference between the total memory and the resident set size (RSS). An error is returned only when the memory has been exceeded. • 1: Most processes use the malloc function to apply for the memory, but do not use all the memory applied. When this parameter is set to 1, the memory requested by the malloc function will be allocated under any circumstances unless there is not sufficient memory. • 2: The swap space is also considered when the system calculates the memory space that can be applied for. You can apply for a large amount of memory even if the swap space is triggered.
overcommit_ratio	The larger the value, the more memory that process can apply for and the less that will be reserved for the operating system. For the formula used to calculate the memory parameters, see Examples to calculate the memory parameters. When this parameter is set to 2, the memory address that can be applied for cannot exceed swap + memory × overcommit_ratio.

The following Database parameters table describes the parameter configuration of the database.

Database parameters

Parameter	Description
gp_vmem_protect_limit	Specifies the maximum amount of memory that all processes can apply for on each node. If the value is too large, it may result in a system OOM error or even more serious problems. If the value is too small, SQL statements may not be executed even when the system has enough memory.

Parameter	Description
runaway_detector_activation_percent	Default value: 90. This value is specified as a percentage. When the memory used by any compute node exceeds runaway_detector_activation_percent × gp_vmem_protect_limit/100, the query is terminated to prevent OOM. The termination starts from the query that occupies the maximum memory until the memory reaches a value lower than runaway_detector_activation_percent × gp_vmem_protect_limit/100. You can use the gp_toolkit.session_level_memory_consumption view to observe the memory usage of each session and runaway information.
statement_mem	Specifies the maximum amount of memory that a single SQL statement can apply. When the maximum memory is exceeded, spill files are created. Default value: 125. Unit: MB. We recommend that you set this parameter according to the following formula: `(gp_vmem_protect_limit × 0.9)/max_expected_concurrent_queries` **Note** • You can specify the statement_mem parameter in a session. If the current concurrency is low and a session needs to run a query that requires a large amount of memory, you must specify this parameter in the session. • Statement_mem is suitable for limiting memory usage in low concurrency scenarios. If you use statement_mem to limit the memory for high concurrency scenarios, each query is allocated with a very small amount of memory. As a result, the performance of a small number of queries with high memory requirements in high concurrency scenarios is affected. We recommend that you use the resource queue to limit the maximum memory usage in high concurrency scenarios.
gp_workfile_limit_files_per_query	Specifies the maximum number of spill files that can be created by each query. When the memory requested by the query exceeds the statement_mem limit, spill files (also known as work files) are created, which is similar to the swap space of the operating system. When the number of spill files used exceeds the limit, the query will be terminated. Default value: 0, which indicates that an unlimited number of spill files can be created.
gp_workfile_compress_algorithm	Specifies the compression algorithm for spill files. Valid values: none and zlib. Specifies the compression algorithm. The values optimize storage space or I/O by sacrificing CPU. You can set this parameter when the disk is insufficient or the spill files meet a write bottleneck.

Examples to calculate the memory parameters

The environment is as follows:

- Host configuration:

```
Total RAM = 256 GB
SWAP = 64 GB
```

- Four hosts, each deployed with eight primary compute nodes and eight secondary compute nodes.

  When a host fails, the eight primary compute nodes are distributed to the remaining three hosts. A single host can be deployed with at most three extra primary compute nodes from the failed host. A single host can be deployed with at most 11 primary compute nodes.

1. Calculate the total memory allocated to AnalyticDB for PostgreSQL by the operating system.

   Reserve 7.5 GB and 5% of memory for the operating system and calculate the available memory for all applications, and divide the available memory by the empirical coefficient of 1.7.

```
gp_vmem = ((SWAP + RAM) - (7.5 GB + 0.05 × RAM))/1.7
        = ((64 + 256) - (7.5 + 0.05 × 256))/1.7
        = 176
```

2. Use the empirical coefficient of 0.026 to calculate overcommit_ratio.

```
vm.overcommit_ratio = (RAM - (0.026 × gp_vmem))/RAM
                    = (256 - (0.026 × 176))/256
                    =. 982
Set vm.overcommit_ratio to 98.
```

3. Calculate gp_vmem_protect_limit (the protection parameter of the maximum memory usage for each compute node), and divide gp_vmem by maximum_acting_primary_segments (the number of primary compute nodes to be run on each other host after one host fails).

```
gp_vmem_protect_limit calculation
gp_vmem_protect_limit = gp_vmem/maximum_acting_primary_segments
                      = 176/11
                      = 16 GB
                      = 16384 MB
```

Configure the resource queue

You can use resource queues to limit the number of concurrent queries and the total memory usage. When a query is running, it is added to the corresponding queue and the resources used are recorded in the queue. The resource limit of the queue is applied to all sessions in the queue.

The resource queue in AnalyticDB for PostgreSQL is similar to cgroup in Linux.

The syntax to create a resource queue is as follows:

```
Command:     CREATE RESOURCE QUEUE
Description: create a new resource queue for workload management
Syntax:
CREATE RESOURCE QUEUE name WITH (queue_attribute=value [, ... ])
where queue_attribute is:
   ACTIVE_STATEMENTS=integer
         [ MAX_COST=float [COST_OVERCOMMIT={TRUE|FALSE}] ]
         [ MIN_COST=float ]
         [ PRIORITY={MIN|LOW|MEDIUM|HIGH|MAX} ]
         [ MEMORY_LIMIT='memory_units' ]
|  MAX_COST=float [ COST_OVERCOMMIT={TRUE|FALSE} ]
         [ ACTIVE_STATEMENTS=integer ]
         [ MIN_COST=float ]
         [ PRIORITY={MIN|LOW|MEDIUM|HIGH|MAX} ]
         [ MEMORY_LIMIT='memory_units' ]
```

The Resource queue creation parameters table describes the parameters for creating the resource queue.

Resource queue creation parameters

Parameter	Description
ACTIVE_STATEMENTS	The number of SQL statements that are allowed to run (in the active state) concurrently. The value -1 indicates an unlimited number of SQL statements can run concurrently.

Parameter	Description

MEMORY_LIMIT 'memory_units kB, MB or GB'	Specifies the maximum memory usage allowed by all SQL statements in the resource queue. The value -1 indicates unlimited memory usage, but it is easy to trigger OOM errors because it is limited by the database or system parameters mentioned in the preceding sections.
	The memory usage of SQL statements is limited by resource queues and parameters.
	• When the gp_resqueue_memory_policy parameter is set to none, the limit is the same as that in the Greenplum databases earlier than version 4.1.
	• When the gp_resqueue_memory_policy parameter is set to auto and you have specified the statement_mem parameter for a session or at the database level, the allowed memory of a single query will exceed the MEMORY_LIMIT of the resource queue.
	Example:
	`=> SET statement_mem='2GB';`
	`=> SELECT * FROM my_big_table WHERE column='value' ORDER BY id;`
	`=> RESET statement_mem;`
	• The system parameter max_statement_mem can limit the maximum memory usage at the compute node level. The memory requested by a single query cannot exceed max_statement_mem.
	You can modify the statement_mem parameter at the session level, but do not modify the max_statement_mem parameter. We recommend that you specify max_statement_mem as follows:
	`(seghost_physical_memory) / (average_number_concurrent_queries)`
	• When the gp_resqueue_memory_policy parameter is set to eager_free, it indicates that the query is divided into several stages and that the database allocates the memory requested in the current stage. For example, if a query requests 1 GB of memory in total but only needs 100 MB during each stage, the database will allocate 100 MB of memory to the query. You can use eager_free to reduce the possibility of insufficient memory for the query.

MAX_COST float	The maximum cost of the queries that are allowed to execute concurrently by the resource group. The cost is the estimated total cost in the SQL execution plan. The value of the parameter can be specified as a floating-point number (such as 100.0) or an exponent (such as 1e+2). A value of -1 indicates the cost is unlimited.
COST_OVERCOMMIT boolean	Specifies whether the limit of max_cost can be exceeded when the system is idle. The value TRUE indicates the limit can be exceeded.
MIN_COST float	When the resources requested exceed the limit, the queries are queued. However, when the cost of a query is lower than the min_cost, the query can run without queuing.
PRIORITY= {MIN|LOW|MEDIUM|HIGH|MAX}	The priority of the current resource queue. When resources are insufficient, CPU resources are allocated to the resource queue with a higher priority. The SQL statements in the resource queue with a higher priority can obtain CPU resources first. We recommend that you allocate users that initiate queries with high real-time requirements to resource queues with higher priority. This parameter is similar to the CPU resource group in the Linux cgroup and the time slice policy of real-time and common tasks.

Example of modifying resource queue limits:

```
ALTER RESOURCE QUEUE myqueue WITH (MAX_COST=-1.0, MIN_COST= -1.0);
```

Example of putting the user in the resource queue:

```
ALTER ROLE sammy RESOURCE QUEUE poweruser;
```

The following table describes the parameters of resource queues.

Resource queue parameters

Parameter	Description
gp_resqueue_memory_policy	Specifies the memory management policy of the resource queue.
gp_resqueue_priority	Specifies whether to enable query prioritization. Valid values: • On. • Off. If this parameter is disabled, existing priority settings are not evaluated.
gp_resqueue_priority_cpucores_per_segment	Specifies the number of CPU cores allocated to each compute node. For example, if an 8-core host is configured with two primary compute nodes, you can set the parameter to 4. If there are no other nodes on the primary node, set the parameter to 8. When the CPU is preempted, the SQL statements running in the resource group with higher priority are allocated with CPU resources first.
gp_resqueue_priority_sweeper_interval	Specifies the interval at which CPU utilization is recalculated for all active statements. The share value is calculated when the SQL statement is executed. You can calculate the share value based on the priority and gp_resqueue_priority_cpucores_per_segment. The smaller the value and the more frequent the calculation, the better the result brought by the priority settings and the larger the overhead.

Tips for configuring resource queues:

- We recommend that you create a resource queue for each user.

  The default resource queue of AnalyticDB for PostgreSQL is pg_default. If no queue is created, all users are assigned to pg_default. This operation is not recommended. We recommend that you create a resource queue for each user. Typically, a database user corresponds to a business. Different database users may correspond to different businesses or users, such as business users, analysts, developers, and DBAs.

- We do not recommend that you use superusers to execute queries.

  Queries initiated by superusers are only limited by the preceding parameters and not by the resource queue. We do not recommend that you use superusers to execute queries if you want to use resource queues to limit the use of resources.

- ACTIVE_STATEMENTS indicates the SQL statements that can be executed concurrently within the resource queue. When the cost of a query is lower than the min_cost, the query can run without queuing.
- You can specify the MEMORY_LIMIT parameter to set the allowed maximum memory usage of all the SQL statements in a resource queue. The statement_mem parameter has higher priority that can break through the limit of resource queues.

> **Note** The memory of all resource queues cannot exceed gp_vmem_protect_limit.

- You can distinguish businesses by configuring the priorities of resource queues.

  For example, assume that report forms have top priority, while common businesses and analysts have lower priorities. In this case, you can create three resource queues with the max, high, and medium priorities, respectively.

- If the number of resources requested at different times vary, you can use the crontab command to adjust the limits of resource queues periodically based on usage patterns.

  For example, the queue of analysts has top priority during the day, while the queue of forms has lower priority at night. AnalyticDB for PostgreSQL does not support resource limits by time period. Therefore, you can only deploy tasks externally by using the ALTER RESOURCE QUEUE statement.

- You can use the view provided by gp_toolkit to observe the resource usage of the resource queues.

```
gp_toolkit.gp_resq_activity
gp_toolkit.gp_resq_activity_by_queue
gp_toolkit.gp_resq_priority_backend
gp_toolkit.gp_resq_priority_statement
gp_toolkit.gp_resq_role
gp_toolkit.gp_resqueue_status
```

15.KVStore for Redis

15.1. User Guide

15.1.1. What is KVStore for Redis?

KVStore for Redis is a key-value storage database service that is compatible with open source Redis protocols. KVStore for Redis supports various data types, such as strings, lists, sets, sorted sets, and hash tables. The service also provides advanced features, such as transactions, message subscription, and message publishing.

You can easily deploy and manage KVStore for Redis databases in the KVStore for Redis console.

- You can create an instance to initialize a database.
- Before you use a KVStore for Redis instance, add IP addresses or CIDR blocks that are used to access the database to the instance whitelist.
- You can manage instances in the KVStore for Redis console.
- To secure data, you can periodically or immediately back up or restore databases in the KVStore for Redis console.
- You can log on to a database by using a client and then execute SQL statements to perform database operations.

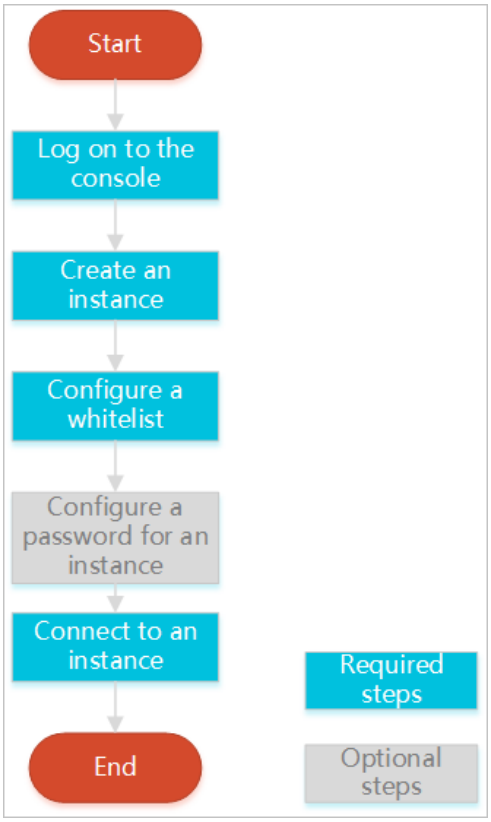
15.1.2. Quick Start

15.1.2.1. Get started with KVStore for Redis

This topic describes a series of operations from creating a KVStore for Redis instance to logging on to a database. This helps you understand the procedures to use KVStore for Redis instances.

The flowchart to use KVStore for Redis instances is as follows.

Flowchart for the KVStore for Redis instance



- Log on to the KVStore for Redis console

  This topic describes how to log on to the KVStore for Redis console.

- Create an instance

  KVStore for Redis supports two network types: classic network and VPC. You can create KVStore for Redis instances of different network types.

- Configure a whitelist

  Before using a KVStore for Redis instance, add IP addresses or CIDR blocks used to access the database to the instance whitelist to improve database security and stability.

- If you have not specified a password when creating the instance, set the password of the instance on the **Instance Information** page.

- Connect to the instance

  You can use a client that supports Redis protocols or use the Redis command-line interface (redis-cli) program to connect to the KVStore for Redis instance.

15.1.2.2. Log on to the KVStore for Redis console

This topic describes how to log on to the KVStore for Redis console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.

2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Login**.
4. In the top navigation bar, choose **Products > KVStore for Redis**.

15.1.2.3. Create an instance

This topic describes how to create an instance in the KVStore for Redis console.

Prerequisites

To use the Virtual Private Cloud (VPC) service, you must create a VPC in the same region as KVStore for Redis.

> **Note** You must specify the network type when you create the instance, and cannot modify the network type later.

Procedure

1. Log on to the KVStore for Redis console.
2. Click **Create Instance** in the upper-right corner.
3. Set the following parameters.

   KVStore for Redis instance parameters

Category	Parameter	Description
Basic Settings	Organization	The organization to which the KVStore for Redis instance to be created belongs.
	Resource Set	The resource set to which the KVStore for Redis instance belongs. **Notice** After you select a resource set, the KVStore for Redis instance is available only to the members of the specified resource set.
Region	Region	Specifies the region where the KVStore for Redis instance is located.
	Zone	Specifies the zone where the KVStore for Redis instance is located.
	Engine Version	The following engine versions are supported: ○ Redis 2.8 ○ Redis 4.0

Category	Parameter	Description
Specifications	Architecture Type	The architecture type of the KVStore for Redis instance. KVStore for Redis provides cluster and standard architectures. The cluster architecture meets large-capacity or high-performance requirements. Native Redis databases run in a single-threading model. If your database does not require high performance, we recommend that you use a standard instance. To achieve higher performance, select a cluster instance.
	Node Type	The node type for the KVStore for Redis instance. KVStore for Redis supports the primary-secondary dual-node structure.
	Instance Class	The specification of the instance. The maximum number of connections and maximum internal bandwidth vary, depending on the instance specification.
Network	Network Type	The network type of the instance. On the Apsara Stack platform, a classic network and a VPC have the following differences: ○ Classic network: Cloud services in a classic network are not isolated. Unauthorized access to a cloud service is blocked only by a security group or a whitelist policy of the service. ○ VPC: A VPC provides an isolated network environment on Apsara Stack. You can customize the routing table, Classless Inter-Domain Routing (CIDR) blocks, and gateway of a VPC. You can also migrate applications to the cloud by using a leased line or virtual private network (VPN) without service interruption to integrate your on-premises data center and cloud resources in a VPC into a virtual data center. **Note** Before you select the VPC type, create a VPC. For more information, see the "Create a VPC" and "Create a VSwitch" topics of *VPC User Guide* .
Set Password	Instance Name	Enter the name of the KVStore for Redis instance. ○ The name must be 2 to 128 characters in length. ○ The name can contain letters, digits, underscores (_), and hyphens (-). It must start with a letter.
	Set Password	You can select **Now** or **Later**.
	Password	Set a password used to connect to the instance. The password must follow these rules: ○ The password must be 8 to 30 characters in length. ○ The password must contain uppercase letters, lowercase letters, and digits. Special characters are not supported.
	Confirm Password	Enter the specified password again.

4. After you set the parameters, click **Submit**.

15.1.2.4. Configure a whitelist

Before using a KVStore for Redis instance, add IP addresses or CIDR blocks used to access the database to the instance whitelist to improve database security and stability.

Context

> **Note** A properly configured whitelist can guarantee the highest-level security protection for your KVStore for Redis instance. We recommend that you maintain the whitelist on a regular basis.

Procedure

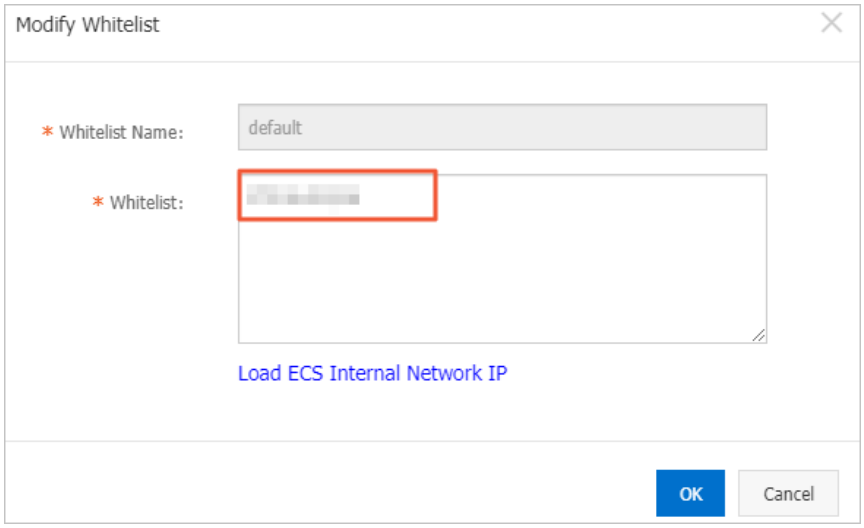
1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. On the **Instance Information** page, click **Whitelist Settings** in the left-side navigation pane.
4. On the **Whitelist Settings** page, proceed in either of the following ways:
   - To customize the whitelist group name, create a new whitelist group:
     a. Click **Add a Whitelist Group** in the upper-right corner.
     b. In the Add a Whitelist Group dialog box that appears, set **Group Name**.

     > **Note** A group name must be 2 to 32 characters in length and contain lowercase letters, digits, or underscores (_). The group name must start with a lowercase letter and end with a letter or digit. You cannot change this name after you create the whitelist group.

   - If you do not require a custom whitelist group, click **Modify** next to the target whitelist group.
5. In the **Add a Whitelist Group** or **Modify Whitelist of Group** dialog box that appears, proceed in either of the following ways:
   - Manually modify the **Whitelist of Group** field:

a. In the **Whitelist of Group** field, enter the IP addresses or CIDR blocks that you can use to connect to the KVStore for Redis instance.

Manually modify the whitelist group



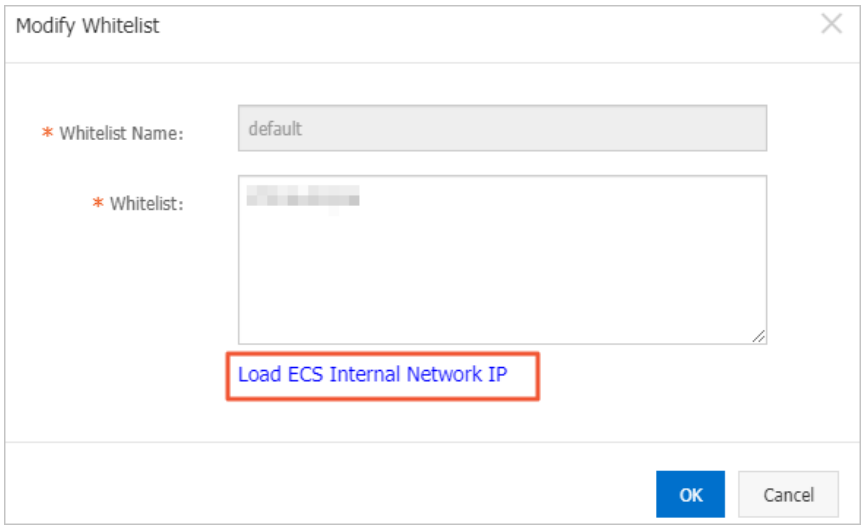
> **Note**
>
> - Set the whitelist to `0.0.0.0/0` to allow connections from all IP addresses.
> - Set the whitelist to `127.0.0.1` to block connections from all IP addresses.
> - Set the whitelist to a CIDR block to allow connections from the IP addresses within the CIDR block, such as `10.10.10.0/24`.
> - When you enter multiple IP addresses or CIDR blocks, separate them with commas (,) and leave no space before or after each comma.
> - You can add 1,000 or fewer IP addresses or CIDR blocks to each whitelist group.

b. Click **OK**.

- Load internal IP addresses of target ECS instances under the current Alibaba Cloud account:

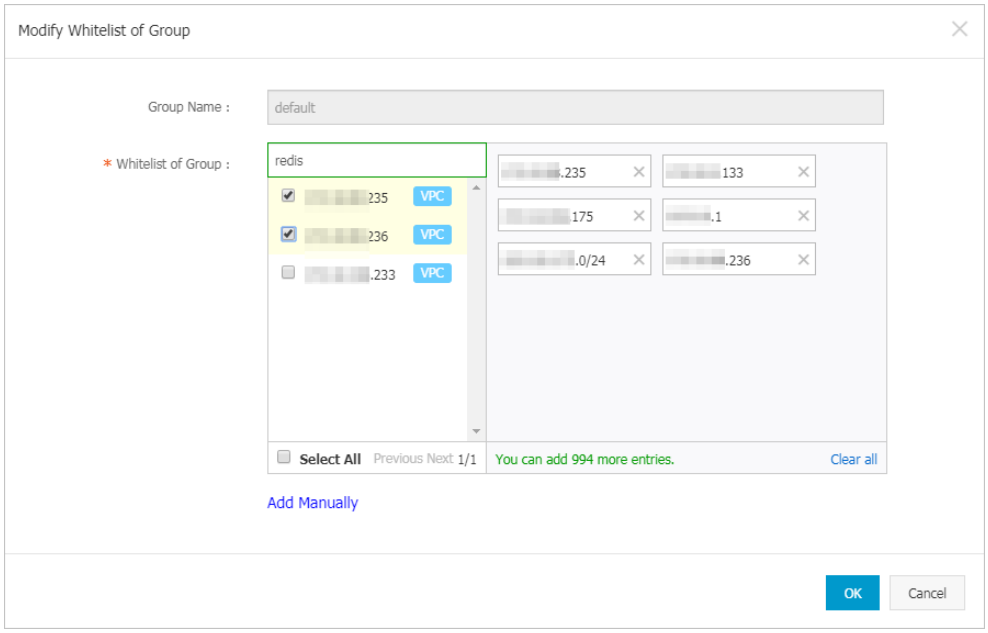
a. Click **Load ECS Internal IP Addresses**.

Load internal IP addresses of target ECS instances



b. Select internal IP addresses of target ECS instances.

Select internal IP addresses of target ECS instances



> **Note** You can perform a fuzzy search by ECS instance name, ID, or IP address on the search bar above the list of ECS internal IP addresses.

c. Click **OK**.

15.1.2.5. Connect to an instance

15.1.2.5.1. Use a Redis client

You can connect to an KVStore for Redis instance by using clients for different programming languages.

The database service of KVStore for Redis is compatible with that of native Redis. Therefore, you can connect to both database services in similar ways. All clients that are compatible with the Redis protocol support connections to KVStore for Redis. You can use any of these clients that are suitable for your applications.

For more information about Redis clients, visit https://redis.io/clients.

Prerequisites

- The internal IP address of the Elastic Compute Service (ECS) instance or the public IP address of the local host has been added to a whitelist of the KVStore for Redis instance. For more information, see Configure a whitelist.
- If you use a custom account to connect to the KVStore for Redis instance, the connection password must be in the format of `<user>:<password>`. For example, if the username of a custom account is `admin` and the password is `password`, the password used to connect to the KVStore for Redis instance must be in the format of `admin:password`.

Jedis clien

You can use a Jedis client to connect to KVStore for Redis in any of the following ways:

- Single Jedis connection. This method is not recommended because a client cannot automatically reconnect to KVStore for Redis after a connection times out.
- JedisPool-based connection. This method is recommended.

To use a Jedis client to connect to an KVStore for Redis instance, perform the following steps:

1. Download and install the Jedis client. For more information, see Jedis.
2. Example of single Jedis connection
   i. Open the Eclipse client, create a project, and then enter the following code:

```
import redis.clients.jedis.Jedis;
public class jedistest {
public static void main(String[] args) {
try {
     String host = "xx.kvstore.aliyuncs.com";//You can view the connection address of the tar
get instance in the console.
     int port = 6379;
     Jedis jedis = new Jedis(host, port);
     //Authentication information.
     jedis.auth("password");//password
     String key = "redis";
     String value = "aliyun-redis";
     //Select a database. Default value: 0.
     jedis.select(1);
     //Set a key.
     jedis.set(key, value);
     System.out.println("Set Key " + key + " Value: " + value);
     //Obtain the configured key and value.
     String getvalue = jedis.get(key);
     System.out.println("Get Key " + key + " ReturnValue: " + getvalue);
     jedis.quit();
     jedis.close();
}
catch (Exception e) {
 e.printStackTrace();
 }
}
}
```

   ii. Run the project. You have connected to KVStore for Redis if you see the following result in the Eclipse console.

```
Set Key redis Value aliyun-redis
Get Key redis ReturnValue aliyun-redis
```

   Then, you can use a Jedis client to manage your KVStore for Redis instance. You can also connect to your KVStore for Redis instance by using JedisPool.

3. Example of JedisPool-based connection
   i. Open the Eclipse client, create a project, and then configure the following pom file:

```
<dependency>
<groupId>redis.clients</groupId>
<artifactId>jedis</artifactId>
<version>2.7.2</version>
<type>jar</type>
<scope>compile</scope>
</dependency>
```

ii. Add the following application to the project:

```
import org.apache.commons.pool2.PooledObject;
import org.apache.commons.pool2.PooledObjectFactory;
import org.apache.commons.pool2.impl.DefaultPooledObject;
import org.apache.commons.pool2.impl.GenericObjectPoolConfig;
import redis.clients.jedis.HostAndPort;
import redis.clients.jedis.Jedis;
import redis.clients.jedis.JedisPool;
import redis.clients.jedis.JedisPoolConfig;
```

iii. If your Jedis client version is Jedis-2.7.2, enter the following code in the project:

```
JedisPoolConfig config = new JedisPoolConfig();
//Maximum number of idle connections. You can customize this parameter. Make sure that the sp
ecified maximum number of idle connections does not exceed the maximum number of connections
that the KVStore for Redis instance supports.
config.setMaxIdle(200);
//Maximum number of connections. You can customize this parameter. Make sure that the specifi
ed maximum number of connections does not exceed the maximum number of connections that the K
VStore for Redis instance supports.
config.setMaxTotal(300);
config.setTestOnBorrow(false);
config.setTestOnReturn(false);
String host = "*.aliyuncs.com";
String password = "Password";
JedisPool pool = new JedisPool(config, host, 6379, 3000, password);
Jedis jedis = null;
try {
jedis = pool.getResource();
/// ... do stuff here ... for example
jedis.set("foo", "bar");
String foobar = jedis.get("foo");
jedis.zadd("sose", 0, "car");
jedis.zadd("sose", 0, "bike");
Set<String> sose = jedis.zrange("sose", 0, -1);
} finally {
if (jedis ! = null) {
jedis.close();
}
}
/// ... when closing your application:
pool.destroy();
```

iv. If your Jedis client version is Jedis-2.6 or Jedis-2.5, enter the following code in the project:

```
JedisPoolConfig config = new JedisPoolConfig();
//Maximum number of idle connections. You can customize this parameter. Make sure that the sp
ecified maximum number of idle connections does not exceed the maximum number of connections
that the KVStore for Redis instance supports.
config.setMaxIdle(200);
//Maximum number of connections. You can customize this parameter. Make sure that the specifi
ed maximum number of connections does not exceed the maximum number of connections that the K
VStore for Redis instance supports.
config.setMaxTotal(300);
config.setTestOnBorrow(false);
config.setTestOnReturn(false);
String host = "*.aliyuncs.com";
String password = "Password";
JedisPool pool = new JedisPool(config, host, 6379, 3000, password);
Jedis jedis = null;
boolean broken = false;
try {
     jedis = pool.getResource();
     /// ... do stuff here ... for example
     jedis.set("foo", "bar");
     String foobar = jedis.get("foo");
     jedis.zadd("sose", 0, "car");
     jedis.zadd("sose", 0, "bike");
     Set<String> sose = jedis.zrange("sose", 0, -1);
}
catch(Exception e)
{
     broken = true;
} finally {
if (broken) {
     pool.returnBrokenResource(jedis);
} else if (jedis ! = null) {
     pool.returnResource(jedis);
 }
}
```

v. Run the project. You have connected to KVStore for Redis if you see the following result in the Eclipse console.

```
Set Key redis Value aliyun-redis
Get Key redis ReturnValue aliyun-redis
```

Then, you can use a Jedis client to manage your KVStore for Redis instance.

PhpRedis client

To use a PhpRedis client to connect to an KVStore for Redis instance, perform the following steps:

1. Download and install the PhpRedis client. For more information, see PhpRedis.
2. In an editor that supports PHP editing, enter the following code:

```
<? php
 /* Replace the following parameter values with the host name and port number of the target instance. */
 $host = "localhost";
 $port = 6379;
 /* Replace the following parameter values with the ID and password of the target instance. */
 $user = "test_username";
 $pwd = "test_password";
 $redis = new Redis();
 if ($redis->connect($host, $port) == false) {
         die($redis->getLastError());
   }
 if ($redis->auth($pwd) == false) {
         die($redis->getLastError());
  }
  /* You can perform database operations after authentication. For more information, visit https://github.com/phpRedis/phpredis. */.
 if ($redis->set("foo", "bar") == false) {
         die($redis->getLastError());
 }
 $value = $redis->get("foo");
 echo $value;
 ? >
```

3. Run the code. Then, you can use a PhpRedis client to connect to your KVStore for Redis instance. For more information, visit https://github.com/phpredis/phpredis.

Redis-py client

To use a redis-py client to connect to an KVStore for Redis instance, perform the following steps:

1. Download and install the redis-py client. For more information, see redis-py.
2. In an editor that supports Python editing, enter the following code. You can use a redis-py client to connect to the KVStore for Redis instance and perform database operations.

```
#! /usr/bin/env python
#-*- coding: utf-8 -*-
import redis
#Replace the following parameter values with the host name and port number of the target instance.
host = 'localhost'
port = 6379
#Replace the following parameter value with the password of the target instance.
pwd = 'test_password'
r = redis.StrictRedis(host=host, port=port, password=pwd)
#You can perform database operations after you establish a connection. For more information, visit https://github.com/andymccurdy/redis-py.
r.set('foo', 'bar');
print r.get('foo')
```

C or C++ client

To use a C or C++ client to connect to an KVStore for Redis instance, perform the following steps:

1. Download, compile, and install the C client by using the following code:

```
    git clone https://github.com/redis/hiredis.git
    cd hiredis
    make
    sudo make install
```

2. Enter the following code in the C or C++ editor:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <hiredis.h>
int main(int argc, char **argv) {
unsigned int j;
redisContext *c;
redisReply *reply;
if (argc < 4) {
        printf("Usage: example xxx.kvstore.aliyuncs.com 6379 instance_id password\n");
        exit(0);
}
const char *hostname = argv[1];
const int port = atoi(argv[2]);
const char *instance_id = argv[3];
const char *password = argv[4];
struct timeval timeout = { 1, 500000 }; // 1.5 seconds
c = redisConnectWithTimeout(hostname, port, timeout);
if (c == NULL || c->err) {
if (c) {
        printf("Connection error: %s\n", c->errstr);
        redisFree(c);
} else {
        printf("Connection error: can't allocate redis context\n");
}
exit(1);
}
/* AUTH */
reply = redisCommand(c, "AUTH %s", password);
printf("AUTH: %s\n", reply->str);
freeReplyObject(reply);
/* PING server */
reply = redisCommand(c,"PING");
printf("PING: %s\n", reply->str);
freeReplyObject(reply);
/* Set a key */
reply = redisCommand(c,"SET %s %s", "foo", "hello world");
printf("SET: %s\n", reply->str);
freeReplyObject(reply);
/* Set a key using binary safe API */
reply = redisCommand(c,"SET %b %b", "bar", (size_t) 3, "hello", (size_t) 5);
printf("SET (binary API): %s\n", reply->str);
freeReplyObject(reply);
/* Try a GET and two INCR */
reply = redisCommand(c,"GET foo");
printf("GET foo: %s\n", reply->str);
freeReplyObject(reply);
reply = redisCommand(c,"INCR counter");
printf("INCR counter: %lld\n", reply->integer);
freeReplyObject(reply);
/* again ... */
reply = redisCommand(c,"INCR counter");
printf("INCR counter: %lld\n", reply->integer);
freeReplyObject(reply);
/* Create a list of numbers, from 0 to 9 */
reply = redisCommand(c,"DEL mylist");
freeReplyObject(reply);
for (j = 0; j < 10; j++) {
```

```
            char buf[64];
            snprintf(buf,64,"%d",j);
            reply = redisCommand(c,"LPUSH mylist element-%s", buf);
            freeReplyObject(reply);
        }
    /* Let's check what we have inside the list */
    reply = redisCommand(c,"LRANGE mylist 0 -1");
    if (reply->type == REDIS_REPLY_ARRAY) {
            for (j = 0; j < reply->elements; j++) {
            printf("%u) %s\n", j, reply->element[j]->str);
    }
    }
    freeReplyObject(reply);
    /* Disconnects and frees the context */
    redisFree(c);
    return 0;
    }
```

3. Compile the code.

```
gcc -o example -g example.c -I /usr/local/include/hiredis -lhiredis
```

4. Run a test.

```
 example xxx.kvstore.aliyuncs.com 6379 instance_id password
```

Now, the C or C++ client is connected to the KVStore for Redis instance.

.NET client

To use a .NET client to connect to an KVStore for Redis instance, perform the following steps:

1. Download and use the .NET client.

```
 git clone https://github.com/ServiceStack/ServiceStack.Redis
```

2. Create a .NET project on the .NET client.
3. Add the reference file stored in the library file directory ServiceStack.Redis/lib/tests to the client.
4. Enter the following code in the .NET project to connect to the KVStore for Redis instance. For more information about API operations, visit https://github.com/ServiceStack/ServiceStack.Redis.

```
using System;
 using System.Collections.Generic;
 using System.Linq;
 using System.Text;
 using System.Threading.Tasks;
 using ServiceStack.Redis;
 namespace ServiceStack.Redis.Tests
 {
        class Program
 {
 public static void RedisClientTest()
 {
        string host = "127.0.0.1";/*IP address of the host that you want to connect to*/
        string password = "password";/*Password*/
        RedisClient redisClient = new RedisClient(host, 6379, password);
        string key = "test-aliyun";
        string value = "test-aliyun-value";
        redisClient.Set(key, value);
        string listKey = "test-aliyun-list";
        System.Console.WriteLine("set key " + key + " value " + value);
```

```
        string getValue = System.Text.Encoding.Default.GetString(redisClient.Get(key));
        System.Console.WriteLine("get key " + getValue);
        System.Console.Read();
 }
 public static void RedisPoolClientTest()
 {
        string[] testReadWriteHosts = new[] {
        "redis://password@127.0.0.1:6379"/*redis://Password@IP address that you want to connect
to:Port*/
 };
 RedisConfig.VerifyMasterConnections = false;//You must set the parameter.
 PooledRedisClientManager redisPoolManager = new PooledRedisClientManager(10/*Number of connectio
ns in the pool*/, 10/*Connection pool timeout value*/, testReadWriteHosts);
 for (int i = 0; i < 100; i++){
        IRedisClient redisClient = redisPoolManager.GetClient();//Obtain the connection.
        RedisNativeClient redisNativeClient = (RedisNativeClient)redisClient;
        redisNativeClient.Client = null;//KVStore for Redis does not support the CLIENT SETNAME
command. Set Client to null.
 try
 {
        string key = "test-aliyun1111";
        string value = "test-aliyun-value1111";
        redisClient.Set(key, value);
        string listKey = "test-aliyun-list";
        redisClient.AddItemToList(listKey, value);
        System.Console.WriteLine("set key " + key + " value " + value);
        string getValue = redisClient.GetValue(key);
        System.Console.WriteLine("get key " + getValue);
        redisClient.Dispose();//
 }catch (Exception e)
 {
        System.Console.WriteLine(e.Message);
 }
 }
        System.Console.Read();
 }
 static void Main(string[] args)
 {
        //Single-connection mode.
        RedisClientTest();
        //Connection-pool mode.
        RedisPoolClientTest();
 }
 }
 }
```

node-redis client

To use a node-redis client to connect to an KVStore for Redis instance, perform the following steps:

1. Download and install a node-redis client.

```
npm install hiredis redis
```

2. Enter and run the following code on the node-redis client to connect to the KVStore for Redis instance.

```
var redis = require("redis"),
client = redis.createClient(<port>, <"host">, {detect_buffers: true});
client.auth("password", redis.print)
```

> **Note** In the code, the port field specifies the port of the KVStore for Redis instance. Default value: 6379. The host field specifies the endpoint of the KVStore for Redis instance. The following example shows the settings of the port and host fields:

```
client = redis.createClient(6379, "r-abcdefg.redis.rds.aliyuncs.com", {detect_buffers: true})
;
```

3. Use the KVStore for Redis instance.

```
  // Write data to the instance.
client.set("key", "OK");
// Query data on the instance. The instance returns data of the String type.
client.get("key", function (err, reply) {
console.log(reply.toString()); // print `OK`
});
// If a buffer is imported, a buffer is returned.
client.get(new Buffer("key"), function (err, reply) {
console.log(reply.toString()); // print `<Buffer 4f 4b>`
});
client.quit();
```

C# client StackExchange.Redis

To use the C# client StackExchange.Redis to connect to an KVStore for Redis instance, perform the following steps:

1. Download and install StackExchange. Redis.
2. Add a reference.

```
using StackExchange.Redis;
```

3. Initialize ConnectionMultiplexer.

   ConnectionMultiplexer is the core of StackExchange.Redis, and shared and reused in the entire application. You must use ConnectionMultiplexer as a singleton. ConnectionMultiplexer is initialized in the following way:

```
 // redis config
 private static ConfigurationOptions configurationOptions = ConfigurationOptions.Parse("127.0.0.1
:6379,password=xxx,connectTimeout=2000");
  //the lock for singleton
 private static readonly object Locker = new object();
  //singleton
 private static ConnectionMultiplexer redisConn;
 //singleton
 public static ConnectionMultiplexer getRedisConn()
 {
     if (redisConn == null)
     {
         lock (Locker)
         {
             if (redisConn == null || ! redisConn.IsConnected)
             {
                 redisConn = ConnectionMultiplexer.Connect(configurationOptions);
             }
         }
     }
     return redisConn;
 }
```

> **Note**
>
> ConfigurationOptions contains multiple options, such as keepAlive, connectRetry, and name. For more information, see StackExchange.Redis.ConfigurationOptions.

4. GetDatabase() returns a lightweight object. You can obtain this object from the object of ConnectionMultiplexer.

```
redisConn = getRedisConn();
var db = redisConn.GetDatabase();
```

5. The following examples show five types of data structures, which are strings, hashes, lists, sets, and sorted sets. The API operations used in these examples are different from their usage in the native Redis service.
   - string

```
//set get
string strKey = "hello";
string strValue = "world";
bool setResult = db.StringSet(strKey, strValue);
Console.WriteLine("set " + strKey + " " + strValue + ", result is " + setResult);
//incr
string counterKey = "counter";
long counterValue = db.StringIncrement(counterKey);
Console.WriteLine("incr " + counterKey + ", result is " + counterValue);
//expire
db.KeyExpire(strKey, new TimeSpan(0, 0, 5));
Thread.Sleep(5 * 1000);
Console.WriteLine("expire " + strKey + ", after 5 seconds, value is " + db.StringGet(strKey));
//mset mget
KeyValuePair<RedisKey, RedisValue> kv1 = new KeyValuePair<RedisKey, RedisValue>("key1", "value
1");
KeyValuePair<RedisKey, RedisValue> kv2 = new KeyValuePair<RedisKey, RedisValue>("key2", "value
2");
db.StringSet(new KeyValuePair<RedisKey, RedisValue>[] {kv1,kv2});
RedisValue[] values = db.StringGet(new RedisKey[] {kv1.Key, kv2.Key});
Console.WriteLine("mget " + kv1.Key.ToString() + " " + kv2.Key.ToString() + ", result is " + v
alues[0] + "&&" + values[1]);
```

   - hash

```
string hashKey = "myhash";
//hset
db.HashSet(hashKey,"f1","v1");
db.HashSet(hashKey,"f2", "v2");
HashEntry[] values = db.HashGetAll(hashKey);
//hgetall
Console.Write("hgetall " + hashKey + ", result is");
for (int i = 0; i < values.Length;i++)
{
  HashEntry hashEntry = values[i];
  Console.Write(" " + hashEntry.Name.ToString() + " " + hashEntry.Value.ToString());
}
Console.WriteLine();
```

   - list

```
//list key
string listKey = "myList";
//rpush
db.ListRightPush(listKey, "a");
db.ListRightPush(listKey, "b");
db.ListRightPush(listKey, "c");
//lrange
RedisValue[] values = db.ListRange(listKey, 0, -1);
Console.Write("lrange " + listKey + " 0 -1, result is ");
for (int i = 0; i < values.Length; i++)
{
 Console.Write(values[i] + " ");
}
Console.WriteLine();
```

- set

```
//set key
string setKey = "mySet";
//sadd
db.SetAdd(setKey, "a");
db.SetAdd(setKey, "b");
db.SetAdd(setKey, "c");
//sismember
bool isContains = db.SetContains(setKey, "a");
Console.WriteLine("set " + setKey + " contains a is " + isContains );
```

- sorted set

```
string sortedSetKey = "myZset";
//sadd
db.SortedSetAdd(sortedSetKey, "xiaoming", 85);
db.SortedSetAdd(sortedSetKey, "xiaohong", 100);
db.SortedSetAdd(sortedSetKey, "xiaofei", 62);
db.SortedSetAdd(sortedSetKey, "xiaotang", 73);
//zrevrangebyscore
RedisValue[] names = db.SortedSetRangeByRank(sortedSetKey, 0, 2, Order.Ascending);
Console.Write("zrevrangebyscore " + sortedSetKey + " 0 2, result is ");
for (int i = 0; i < names.Length; i++)
{
  Console.Write(names[i] + " ");
}
Console.WriteLine();
```

15.1.2.5.2. Use redis-cli

You can use the Redis command-line interface (redis-cli) tool to connect to a KVStore for Redis instance.

Notice Only connections over an internal network are supported by KVStore for Redis. Therefore, only the Elastic Compute Service (ECS) instances that run in the same Virtual Private Cloud (VPC) network as KVStore for Redis and that are installed with redis-cli can connect to KVStore for Redis for data management.

Install the redis-cli utility

Install a Linux-based Redis software distribution that includes the redis-cli utility. For more information about the detailed procedure, see Redis community.

Prerequisites

Connection over an internal network

- If the KVStore for Redis instance that you want to manage and the ECS instance that connects to the KVStore for Redis instance run in a classic network, both instances must be located in the same region.
- You have added the internal IP address of the ECS instance to an IP address whitelist of the KVStore for Redis instance.
- The operating system of the local host must be Linux.
- You have installed the Redis software distribution on the ECS instance.
-

Connection over the Internet

- You have applied for a public endpoint for the KVStore for Redis instance.
- You have added the public IP address of the local host to an IP address whitelist of the KVStore for Redis instance.
- The operating system of the local host must be Linux.
- You have installed the Redis software distribution on the local host.
-

Connect to a KVStore for Redis instance

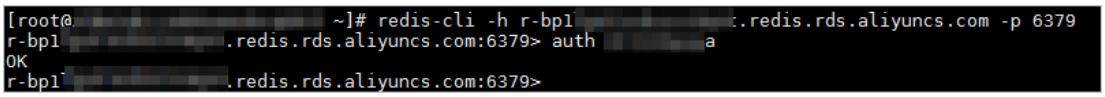
On the command line, run the following command to connect to a KVStore for Redis instance.

```
redis-cli -h <host> -p <port> -a <password>
```

Parameters

Parameter	Description
-h	The endpoint of the KVStore for Redis instance.
-p	The service port of the KVStore for Redis instance. The default port number is 6379 and cannot be changed.
-a	The password used to connect to the KVStore for Redis instance. You can skip this parameter to avoid revealing the password in plaintext and enhance security. After you run the preceding command, you can enter `auth <password>` to complete the authentication. The following figure shows an example.

Sample code



15.1.3. Instance management

15.1.3.1. Change the password

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.

4.

15.1.3.2. Configure a whitelist

Before using a KVStore for Redis instance, add IP addresses or CIDR blocks used to access the database to the instance whitelist to improve database security and stability.

Context

> **Note** A properly configured whitelist can guarantee the highest-level security protection for your KVStore for Redis instance. We recommend that you maintain the whitelist on a regular basis.

Procedure

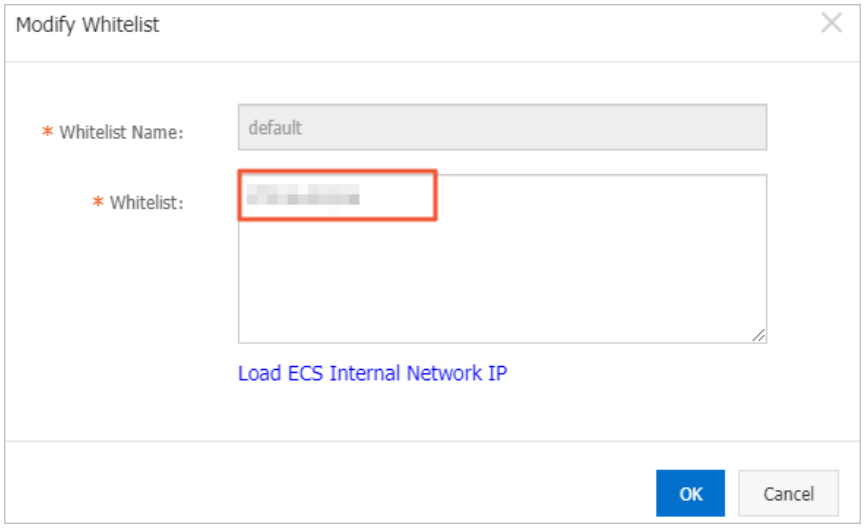
1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. On the **Instance Information** page, click **Whitelist Settings** in the left-side navigation pane.
4. On the **Whitelist Settings** page, proceed in either of the following ways:
   - To customize the whitelist group name, create a new whitelist group:
     a. Click **Add a Whitelist Group** in the upper-right corner.
     b. In the Add a Whitelist Group dialog box that appears, set **Group Name**.

     > **Note** A group name must be 2 to 32 characters in length and contain lowercase letters, digits, or underscores (_). The group name must start with a lowercase letter and end with a letter or digit. You cannot change this name after you create the whitelist group.

   - If you do not require a custom whitelist group, click **Modify** next to the target whitelist group.
5. In the **Add a Whitelist Group** or **Modify Whitelist of Group** dialog box that appears, proceed in either of the following ways:
   - Manually modify the **Whitelist of Group** field:

a. In the **Whitelist of Group** field, enter the IP addresses or CIDR blocks that you can use to connect to the KVStore for Redis instance.

Manually modify the whitelist group



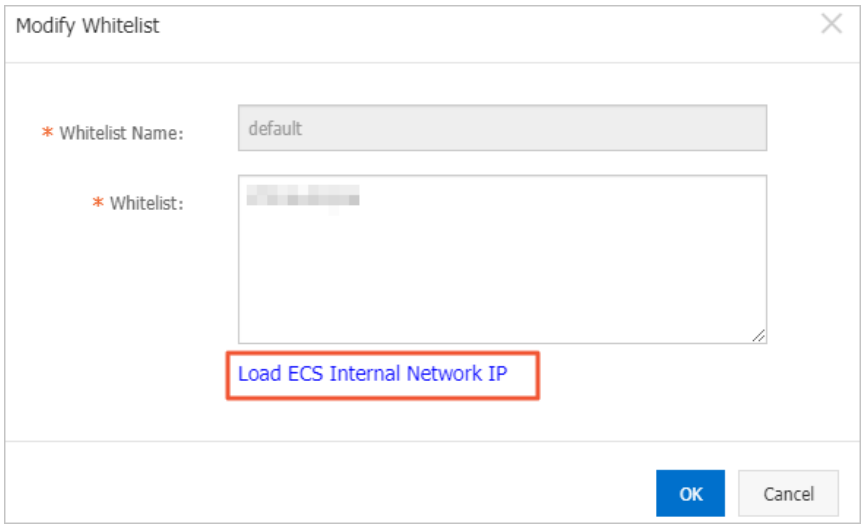
> **Note**
>
> - Set the whitelist to `0.0.0.0/0` to allow connections from all IP addresses.
> - Set the whitelist to `127.0.0.1` to block connections from all IP addresses.
> - Set the whitelist to a CIDR block to allow connections from the IP addresses within the CIDR block, such as `10.10.10.0/24`.
> - When you enter multiple IP addresses or CIDR blocks, separate them with commas (,) and leave no space before or after each comma.
> - You can add 1,000 or fewer IP addresses or CIDR blocks to each whitelist group.

b. Click **OK**.

- Load internal IP addresses of target ECS instances under the current Alibaba Cloud account:

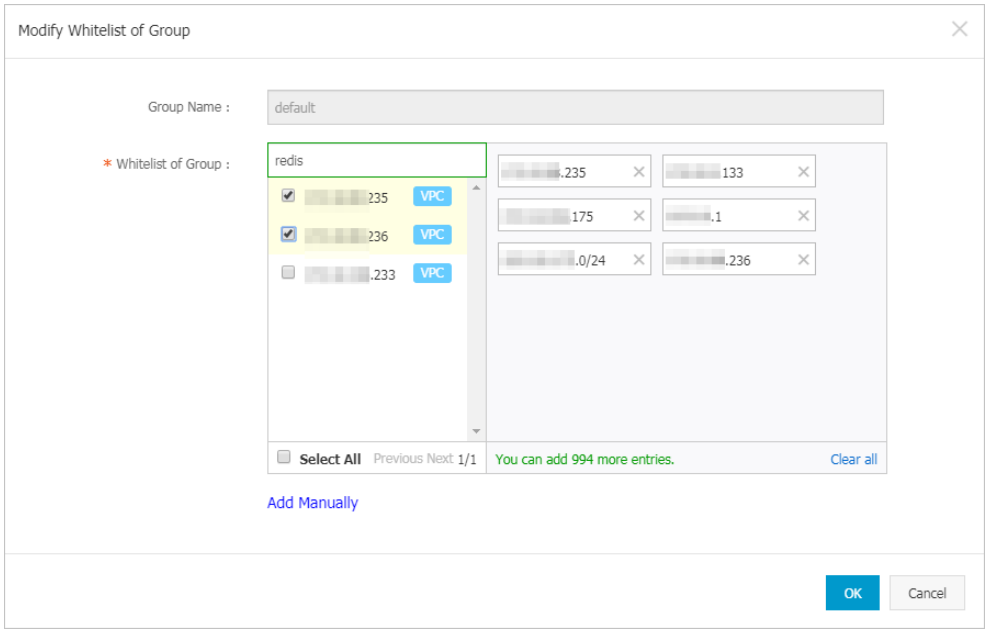
a. Click **Load ECS Internal IP Addresses**.

Load internal IP addresses of target ECS instances



b. Select internal IP addresses of target ECS instances.

Select internal IP addresses of target ECS instances



Note You can perform a fuzzy search by ECS instance name, ID, or IP address on the search bar above the list of ECS internal IP addresses.

c. Click **OK**.

15.1.3.3. Change configurations

Context

Procedure

1. Log on to the KVStore for Redis console.
2.
3. On the **Change Configurations** page, change the configurations and click **Submit**.

   The following example provides common configurations:

Configuration	Description
Architecture Type	The architecture type of the KVStore for Redis instance. KVStore for Redis provides cluster and standard architectures. The cluster architecture meets large-capacity or high-performance requirements. Native Redis databases run in a single-threading model. If your database does not require high performance, we recommend that you use a standard instance. To achieve higher performance, select a cluster instance.
Instance Class	The specification of the instance. The maximum number of connections and maximum internal bandwidth vary, depending on the instance specification.

15.1.3.4. Set a maintenance window

Context

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4. Select time periods and click **Save**.

> **Note** The time periods are in UTC+8.

15.1.3.5. Upgrade the minor version

Context

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4.

15.1.3.6. Configure SSL encryption

Context

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4.
5. In the **Configure SSL** dialog box that appears, turn on the **Enable** switch. The switch will turn from green to gray when it is enabled. Click **OK**.
   - If an error message is displayed to indicate that the instance is in an abnormal state, click OK in the message that appears.
   - If an error message is displayed to indicate that the feature is not supported in this version, upgrade the minor version of the instance. For more information, see Upgrade the minor version.
   - After the operation, you must wait for a short period of time before the system displays the operation result.
   - You can also click **Update Validity** and **Download CA Certificate** in the upper-right corner of the SSL Settings page to perform relevant operations.

15.1.3.7. Clear data

Context

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4.

15.1.3.8. Release an instance

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4.

15.1.3.9. Manage database accounts

KVStore for Redis allows you to create up to 20 database accounts for an instance. You can grant permissions to these accounts and manage your instance based on the actual needs to minimize misoperations.

Prerequisites

The engine version of the instance is Redis 4.0 or later.

> **Note** If the engine version of the instance is not Redis 4.0, only the default account is available. The default account is created when you create the instance. For more information about how to change the password of the default account, see Change the password.

Context

Create an account

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3. On the Instances page, find the instance that you want to manage and click the instance ID. By default, the **Instance Information** page is displayed. In the left-side navigation pane, click **Account Management**.

   > **Note** If Account Management is not available for an instance of Redis 4.0 or later, you can try to upgrade the minor version. For more information, see Upgrade the minor version.

4.
5. In the **Create Account** dialog box that appears, set the following parameters and click **OK**.

   Set account parameters

Parameter	Description
Account	Your account must meet the following requirements: ○ The account name can contain lowercase letters, digits, underscores (_), and hyphens (-). ○ The name must start with a lowercase letter. ○ The name must be 1 to 16 characters in length.

Parameter	Description
Privilege	The permissions granted to the account. Valid values: Read-only, Read/Write, and Replicate. If you select Replicate, you are authorized to use the SYNC and PSYNC commands after you connect to an instance with this account. **Note** You can create accounts that have the replicate permission only for standard instances of Redis 4.0 or later.
Password Settings	The password of your account must meet the following requirements: ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ The password must be 8 to 32 characters in length. Special characters include: ! @#$%^&*()+-=_
Confirm Password	Enter the password again.
Description	The description of an account must meet the following requirements: ○ The description can contain letters, Chinese characters, digits, underscores (_), and hyphens (-). ○ The description must start with a letter or Chinese character, ○ and cannot start with *http://* or *https://*. ○ The description must be 2 to 256 characters in length. You can skip this parameter when you create an account.

Note After you create an account, the account is in the **Unavailable** status. After about one minute, its status changes to **Available**.

15.1.3.10. Use a Lua script

Support for Lua commands

Note If the Eval command cannot be executed, such as when the "`ERR command eval not support for normal user`" message is displayed, you can try to Upgrade the minor version. During the upgrade, the instance may be disconnected and become read-only for a few seconds. We recommend that you upgrade the version of an instance during off-peak hours.

15.1.3.11. Restart an instance

Procedure

1. Log on to the KVStore for Redis console.
2.
3. In the dialog box that appears, select a restart time and click **OK**.
   ○ Restart Immediately: restarts the instance immediately.

- Restart Within Maintenance Window: restarts the instance within the preset maintenance window.

15.1.3.12. Export the list of instances

Procedure

1. Log on to the KVStore for Redis console.
2.
3.

15.1.4. Connection management

15.1.4.1. View connection strings

Context

> Note
> - The virtual IP address of a KVStore for Redis instance may change when you maintain or modify the service. To ensure connection availability, we recommend that you use a connection string to access the KVStore for Redis instance.
> - For more information about how to apply for a public connection string, see Applies for a public connection string.

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.

15.1.4.2. Apply for a public endpoint

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4. In the **Apply for External IP Address** dialog box that appears, enter an endpoint and port number, and click **OK**.

   > Note
   > - The custom endpoint prefix must be 8 to 64 characters in length and can contain lowercase letters and digits. It must start with a lowercase letter.
   > - The custom port ranges from 1024 to 65535. The default value is 6379.
   > - After you apply for a public endpoint, you must add the public IP address to an IP address whitelist of the instance to connect to the instance over the Internet. For more information, see Configure a whitelist.

5.

15.1.4.3. Modify the endpoint of an KVStore for Redis instance

Prerequisites

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4. In the **Modify Public Endpoint** dialog box, set the following parameters:

Parameter	Description
Connection type	Select **Internal Endpoint** or **Public Endpoint**.
Endpoint	Set the prefix of the endpoint. ○ The endpoint can contain lowercase letters and digits. ○ It must start with a lowercase letter. ○ The endpoint must be 8 to 64 characters in length.
Port	Specify a port number. Valid value: 1024 to 65535. **Note** It takes about 10 minutes for the modified port number of the public endpoint to take effect. You can refresh the page to view the latest port number information.

5. In the **Modify Public Endpoint** dialog box, modify **Connection Type**, **Endpoint**, and **Port**, and then click **OK**.

> **Note**
> ○ The custom endpoint prefix must be 8 to 64 characters in length and can contain lowercase letters and digits. It must start with a lowercase letter.
> ○ The custom port number ranges from 1024 to 65535. The default value is 6379.

15.1.5. Parameter configuration

Context

Configure parameters in the KVStore for Redis console

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4.
5.

15.1.6. Backup and recovery

15.1.6.1. Back up data automatically

An increasing number of applications use KVStore for Redis for persistent storage. Because of this, KVStore for Redis supports routine backup mechanisms to restore data after misoperations occur. Alibaba Cloud provides secondary nodes to back up .rdb files as snapshots. Backup operations do not affect the performance of your instance. You can customize the backup operation in the console.

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4.
5.
6.

15.1.6.2. Back up data manually

You can initiate a manual backup task in the console at any time.

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4.
5.

15.1.6.3. Download backup files

To archive these backup files for a longer period, you can copy their URLs in the console and download the database backup files to a local directory.

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4.
5. In the **Download Backup File** dialog box that appears, click one of the following buttons to continue with the procedure:
   - **Download**: downloads the backup file to a local directory.
   - **Get URL for Internet**: copies the public URL for downloading the backup file, and downloads the backup file over the Internet.
   - **Get URL for Intranet**: copies the internal URL for downloading the backup file, and downloads the backup file over the internal network.
   - **Cancel**: cancels downloading the backup file.

15.1.6.4. Restore data

You can use backup files to restore data in the console.

Context

Notice

- Data restoration is highly risky. Check the data to be restored before performing this operation. Proceed with caution.
- This feature is not applicable to non-cluster KVStore for Redis instances.

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4.
5. In the **Restore Data** dialog box that appears, click **Continue**.

   You can apply backup files to a new instance by cloning an instance.

15.1.6.5. Clone an instance

You can apply backup files to a new instance by cloning an instance.

Context

Note This feature is applicable to non-cluster KVStore for Redis instances.

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4.
5.

15.1.7. Performance monitoring

15.1.7.1. View monitoring data

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4.
5. Set a time range in which you want to query monitoring data and click **OK**.

   Note For more information about the monitoring metrics, see Understand metrics.

15.1.7.2. Customize metrics

Context

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4.
5.

15.1.7.3. Modify monitoring frequency

Context

Procedure

1. Log on to the KVStore for Redis console.
2. On the **Instance List** page, click the ID of the instance.
3.
4.
5.

15.1.7.4. Understand metrics

KVStore for Redis supports more than 10 group of metrics for real-time monitoring. You can use these metrics to monitor the running status of your KVStore for Redis instances. This topic describes these metrics.

Basic monitoring metrics

Metric	Unit	Description	Statistical method
CpuUsage	%	The CPU usage.	Monitor the CPU usage when collecting monitoring data.
UsedMemory	Bytes	The amount of the used memory.	Monitor the amount of the used memory when collecting monitoring data.
TotalQps	Counts/s	The number of requests received by an instance per second.	Divide the number of requests in a monitoring cycle by the number of seconds in the monitoring cycle.
ConnCount	Counts	The number of connections.	Monitor the number of connections when collecting monitoring data.

Metric	Unit	Description	Statistical method
InFlow	Kbit/s	The amount of data received by an instance per second.	Divide the amount of data received in a monitoring cycle by the number of seconds in the monitoring cycle.
OutFlow	Kbit/s	The amount of data sent by an instance per second.	Divide the amount of data sent in a monitoring cycle by the number of seconds in the monitoring cycle.
FailedCount	Counts/s	The average number of abnormal requests per second.	Divide the total number of abnormal requests in a monitoring cycle by the number of seconds in the monitoring cycle.
AvgRt	μs	The average response time of all requests. **Note** For more information, see Response time metrics.	Divide the processing time of all requests in a monitoring cycle by the number of requests in the monitoring cycle.
MaxRt	μs	The maximum response time of requests. **Note** For more information, see Response time metrics.	Monitor the maximum amount of time consumed for processing a single request in a monitoring cycle.
Keys	Counts	The total number of keys.	Monitor the number of keys when collecting monitoring data.
Expires	Counts	The total number of keys for which an expiration time is set.	Monitor the total number of keys for which an expiration time is set when collecting monitoring data.
ExpiredKeys	Counts	The total number of expired keys.	Monitor the cumulative sum when collecting monitoring data. After the instance is restarted, the cumulative sum is calculated again.

Metric	Unit	Description	Statistical method
EvictedKeys	Counts	The total number of keys that are evicted because the memory is fully occupied.	Monitor the cumulative sum when collecting monitoring data. After the instance is restarted, the cumulative sum is calculated again.
request	Bytes	The total amount of request data received by KVStore for Redis nodes in a monitoring cycle.	See the description of this metric.
response	Bytes	The total amount of response data sent by KVStore for Redis nodes in a monitoring cycle.	See the description of this metric.
request_max	Bytes	The maximum amount of data that a single request has in a monitoring cycle.	See the description of this metric.
response_max	Bytes	The maximum amount of data that a single response has in a monitoring cycle.	See the description of this metric.
traffic_control_input	Counts	The number of times that downstream throttling is triggered.	Monitor the cumulative sum in a monitoring cycle.
traffic_control_output	Counts	The number of times that upstream throttling is triggered.	Monitor the cumulative sum in a monitoring cycle.
traffic_control_input_status	Counts	Indicates whether downstream throttling has been triggered in a monitoring cycle. A value of 0 indicates that throttling has not been triggered, and a value of 1 indicates that throttling has been triggered.	See the description of this metric.
traffic_control_output_status	Counts	Indicates whether upstream throttling has been triggered in a monitoring cycle. A value of 0 indicates that throttling has not been triggered, and a value of 1 indicates that throttling has been triggered.	See the description of this metric.

Metric	Unit	Description	Statistical method
hit_rate	%	The request hit ratio. This metric indicates the probability that data exists in a KVStore for Redis instance when the data is requested.	Calculate the percentage of the hit requests to the total number of requests in a monitoring cycle.
hit	Counts	The number of hit requests.	Monitor the number of hit requests in a monitoring cycle.
miss	Counts	The number of missed requests.	Monitor the number of missed requests in a monitoring cycle.
evicted_keys_per_sec	Counts/s	The number of keys that are evicted per second.	Divide the total number of keys evicted in a monitoring cycle by the number of seconds in the monitoring cycle.

Other monitoring metrics

Besides the basic monitoring metrics, the system uses other monitoring metrics to monitor specific types of data or specific features. These monitoring metrics are classified into:

- Metrics that reflect the number of times that commands are used. For example, the del, dump, and exists metrics for keys monitoring indicate the number of times the DEL, DUMP, and EXISTS commands are used.
- Response time metrics of commands. For example, the metrics that are ended with avg_rt for keys monitoring, such as del_avg_rt, dump_avg_rt, and exists_avg_rt indicate the average response time of the DEL, DUMP, and EXISTS commands in a monitoring cycle.

Response time metrics

All groups of monitoring metrics include response time metrics. Such metrics are ended with Rt or rt, for example, the AvgRt and MaxRt metrics in the basic monitoring metrics or the del_avg_rt and exists_avg_rt metrics in the keys monitoring.

The AvgRt and MaxRt metrics in the basic monitoring group are the most frequently-used response time metrics. These metrics have different meanings for proxy nodes and data nodes.

- For a cluster instance or a read/write splitting instance, the AvgRt metric of a proxy node reflects the average time consumed by the proxy node to process a command. A proxy node processes a command by following these steps:
  i. The proxy node receives a command and forwards the command to a data node.
  ii. The data node processes the command and responds to the proxy node.
  iii. The proxy node returns the command processing result.

  The AvgRt metric of the proxy node includes the amount time consumed by the data node to process a command, the waiting time, and the amount of time consumed for network communication between the proxy node and the data node.
- For data nodes of a cluster instance or a read/write splitting instance or for a standard instance, the AvgRt metric reflects the average time consumed by a data node to process a command. This metric records the period from the time when the data node receives the command to the time when the data node returns the processing result. This metric does not include the time consumed by the proxy node to process the command and the time consumed for network communication.

- The MaxRt metric indicates the maximum response time of requests. The statistical method of this metric is similar to that of the AvgRt metric for all KVStore for Redis instances.

16.ApsaraDB for MongoDB

16.1. User Guide

16.1.1. Usage notes

You must get familiar with the precautions and limits of ApsaraDB for MongoDB before you start.

To ensure the stability and security of ApsaraDB for MongoDB instances, take note of the limits, see Instance types in *ApsaraDB for MongoDB ApsaraDB for MongoDB limits*.

ApsaraDB for MongoDB limits

Item	Limit
Scale out nodes	You cannot scale out secondary nodes.
	• When a replica set instance is created, three nodes are added. • ApsaraDB for MongoDB provides a primary node, a secondary node, and a hidden node for each replica set instance. The hidden node is invisible to you. • You cannot scale out secondary nodes.
Restart an instance	You must restart an ApsaraDB for MongoDB instance in the ApsaraDB for MongoDB console or by calling the API operation.

16.1.2. Log on to the ApsaraDB for MongoDB console

This topic describes how to log on to the ApsaraDB for MongoDB console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

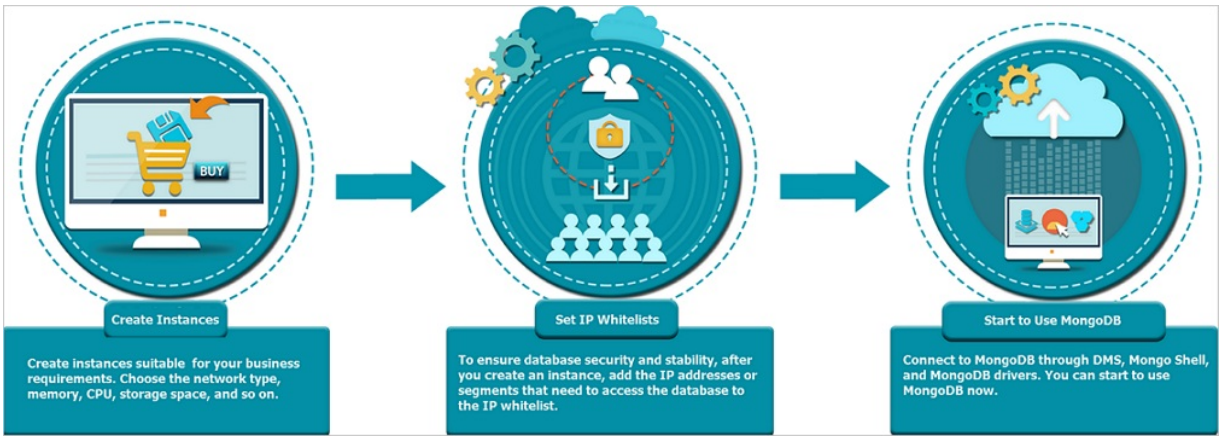
   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products** > **Database Services** > **ApsaraDB for MongoDB**.

16.1.3. Quick start

16.1.3.1. Use ApsaraDB for MongoDB

This topic is a quick start guide to basic usage operations for ApsaraDB for MongoDB, such as creating an instance, configuring a whitelist, and connecting to an instance. Flowcharts are used to describe the basic procedures in ApsaraDB for MongoDB, and guide you to create an ApsaraDB for MongoDB instance.



- Create an ApsaraDB for MongoDB instance

  An instance is a virtual database server on which you can create and manage multiple databases.

- Configure a whitelist for an ApsaraDB for MongoDB instance

  After you create an ApsaraDB for MongoDB instance, you need to configure a whitelist for the instance to allow external devices to access the instance.

  A whitelist can enhance access security for ApsaraDB for MongoDB instances. We recommend that you update the whitelist on a regular basis. The normal services of the instance are not affected if you configure a whitelist.

- Connect to a replica set instance by using the mongo shell

  After you create an instance and configure a whitelist, you can use the mongo shell to connect to the instance.

16.1.3.2. Create an ApsaraDB for MongoDB instance

This topic describes how to create an instance in the ApsaraDB for MongoDB console.

Prerequisites

An account is obtained to log on to the ApsaraDB for MongoDB console.

Create a replica set instance

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. On the **Replica Set Instances** page, click **Create Instance** in the upper-left corner. On the **Create ApsaraDB for MongoDB Instance** page, configure the parameters.

   The following table describes the required parameters.

   Parameters for creating a replica set instance

Section	Parameter	Description
Basic Settings	**Organization**	Select an organization for the new instance.
	Resource Set	Select a resource set for the new instance.
Region	**Region**	Select a region for the new instance.
	Zone	Select a zone for the new instance
Specifications	**Database Engine**	Select a database engine for the new instance. In this case, you can select only **MongoDB**.
	Engine Version	Select a database engine version for the new instance. Valid values: ○ 3.0 ○ 3.4 ○ 4.0 ○ 4.2
	Node Type	ApsaraDB for MongoDB supports the following options: ○ Three-node Replica Set: uses dedicated memory and I/O resources but shares CPU and storage resources with other general-purpose instances on the same server. ○ Dedicated Instance: uses dedicated CPU, memory, storage, and I/O resources to ensure long-term performance. In this case, an instance is not affected by other instances on the same server. ○ Dedicated Host: exclusively uses all resources of a server. This is the top configuration of exclusive specifications.
	Node Specifications	Select a node specification for the new instance. For more information, see descriptions in the ApsaraDB for MongoDB console.
	Storage Capacity (GB)	The storage space of the instance, which contains the space for data, system files, log files, and transaction files. For more information, see *Instance types* in *ApsaraDB for MongoDB Product Introduction*.
Network	**Network Type**	ApsaraDB for MongoDB supports the following options: ○ **Classic Network**: Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by security groups or whitelists. ○ **VPC**: Virtual Private Cloud (VPC) is an isolated network environment built in Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for improved security. **Note** If you select the **VPC** network type, you must configure the **VPC** and **vSwitch** parameters.
	VPC	Select a VPC. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.

Section	Parameter	Description
	vSwitch	Select a vSwitch. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.
Password Settings	**Instance Name**	Set the name of the new instance. The name must be 2 to 256 characters in length, The name must start with a letter, and can contain digits, letters, underscores (_), and hyphens (-).
	Password Setting	Determine when to set the password for logging on to databases in the new instance. You can select **Set Now** to set the logon password immediately, or select **Set after Purchase** to set the logon password after you create the instance. For more information, see Reset the password for an ApsaraDB for MongoDB instance.
	Logon Password	Set a password. The password must meet the following requirements: ○ It must be 8 to 32 characters in length. ○ It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ The name can contain special characters. Special characters include ! # $ % ^ & * ( ) _ + =
	Confirm Password	Enter the password again. The password you enter here must be the same as that in New Password.

3. Click **Submit** to create the instance.

Create a sharded cluster instance

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.
3. Click **Create Instance** in the upper-left corner to go to the **Create MongoDB Instance** page. Configure the parameters.

   The following table describes the required parameters.

   Parameters for creating a sharded cluster instance

Section	Parameter	MNS logs
Basic Settings	**Organization**	Select an organization for the new instance.
	Resource Set	Select a resource set for the new instance.
	Region	Select a region for the new instance.

Section	Parameter	MNS logs
Region	Zone	Select a zone for the new instance
Specifications	Database Engine	Select a database engine for the new instance. In this case, you can select only **MongoDB**.
	Engine Version	Select a database engine version for the new instance. Valid values: ○ 3.4 ○ 4.0 ○ 4.2
Network	Network Type	ApsaraDB for MongoDB supports the following options: ○ **Classic Network**: Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by security groups or whitelists. ○ **VPC**: Virtual Private Cloud (VPC) is an isolated network environment built in Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for improved security. **Note** If you select the **VPC** network type, you must configure the **VPC** and **vSwitch** parameters.
	VPC	Select a VPC. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.
	vSwitch	Select a vSwitch. If no vSwitch exists, you can click **Create vSwitch** to create a vSwitch. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.
Mongos Specifications	Mongos Specifications	The specifications of the mongos node. For more information, see descriptions in the ApsaraDB for MongoDB console.
	Quantity	The number of the mongos nodes. You can select 2 to 32 mongos nodes.
Shard Specifications	Shard Specifications	The specifications of the shard node. For more information, see descriptions in the ApsaraDB for MongoDB console.
	Storage Capacity (GB)	The storage space of the instance, which contains the space for data, system files, log files, and transaction files. For more information, see *Instance types* in *ApsaraDB for MongoDB Product Introduction*.
	Quantity	The number of the shard nodes. You can select 2 to 32 shard nodes.
Config Server	Config Server Specifications	The specifications of Configserver nodes. It is fixed at **1 vCPU, 2 GiB memory** and cannot be customized.

Section	Parameter	MNS logs
Specifications	Storage Capacity (GB)	The storage space of the Configserver node. It is fixed at 20 GB and cannot be customized.
Password Settings	Instance Name	Set the name of the new instance. The name must be 2 to 256 characters in length, and can contain digits, letters, underscores (_), and hyphens (-). It must start with a letter.
	Password Setting	Determine when to set the password for logging on to databases in the new instance. You can select **Set Now** to set the logon password immediately, or select **Set after Purchase** to set the logon password after you create the instance. For more information, see Reset the password for an ApsaraDB for MongoDB instance.
	Logon Password	Set a password. The password must meet the following requirements: ○ It must be 8 to 32 characters in length. ○ It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ The name can contain special characters. Special characters include ! # $ % ^ & * ( ) _ + =
	Confirm Password	Enter the password again. The password you enter here must be the same as that in New Password.

4. Click **Submit** to create the instance.

16.1.3.3. Configure a whitelist for an ApsaraDB for MongoDB instance

This topic describes how to configure a whitelist for an ApsaraDB for MongoDB instance. Before you use an ApsaraDB for MongoDB instance, you must add the IP addresses or Classless Inter-Domain Routing (CIDR) blocks that you use for database access to a whitelist of this instance. This improves database security and stability. Proper configuration of whitelists can enhance access security of ApsaraDB for MongoDB. We recommend that you maintain the whitelists on a regular basis.

Context

The system creates a default whitelist for each instance. This whitelist can be modified or cleared, but it cannot be deleted. After an ApsaraDB for MongoDB instance is created, the system automatically adds the IP address 127.0.0.1 to the **default** whitelist of this instance. The IP address 0.0.0.0/0 indicates that all IP addresses are allowed to access this instance. You must add the IP addresses or CIDR blocks that you allow to access this ApsaraDB for MongoDB instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, choose **Data Security > Whitelist Settings**.

5. You can manually configure a whitelist or import ECS Internal IP addresses to the whitelist.

   Manually modify a whitelist

   i. Find the whitelist you want to modify and choose > **Manually Modify** in the **Actions** column.

   ii. Enter IP addresses or CIDR blocks.

   > Note
   > - Separate multiple IP addresses with commas (,). You can add a maximum of 1,000 different IP addresses to a whitelist. Supported formats are IP addresses such as 0.0.0.0/0 and 10.23.12.24, or CIDR blocks such as 10.23.12.24/24. /24 indicates the length of the IP address prefix in the CIDR block. An IP address prefix can contain 1 to 32 bits.
   > - If the IP whitelist is empty or only contains `0.0.0.0/0`, all devices are granted access. This is risky for your ApsaraDB for MongoDB instance. We recommend that you add only the IP addresses or CIDR blocks of your own web servers to the whitelist.

   iii. Click **OK**.

   Load IP addresses of ECS instances

   i. Find the whitelist, and choose > **Import ECS Intranet IP** in the **Actions** column.

   ii. From the displayed internal IP addresses of ECS instances that belong to the current account, select the IP addresses and click > to add them to the whitelist.

   iii. Click **OK**.

16.1.3.4. Overview of replica set instance connections

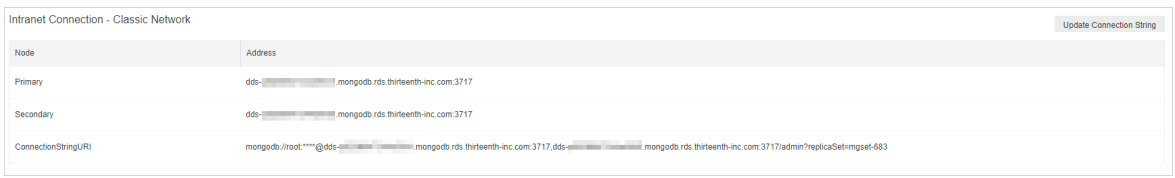
ApsaraDB for MongoDB supports both connection strings and connection string URIs. You can use a connection string to connect to either the primary or secondary node, and use a connection string URI to connect to both of them. For high availability, we recommend that you use connection string URIs to connect your application to both primary and secondary nodes. This topic provides an overview of replica set instance connections.

Prerequisites

A whitelist is configured for the instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

View connection strings

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. On the **Replica Set Instances** page, click the ID of an instance, or click in the **Actions** column corresponding to the instance and select **Manage**.
3. In the left-side navigation pane, click **Database Connection** to view connection strings.



Description of connection strings

Item	Description
Type	• Intranet Connection - Classic Network: Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by security groups or whitelists. • Intranet Connection - VPC: A VPC is an isolated network with higher security and performance than a classic network. By default, ApsaraDB for MongoDB provides endpoints on a VPC.
Role	• Primary: the primary node in the replica set instance. If you connect to this node, you can perform read and write operations on the databases of the replica set instance. • Secondary: the secondary node in the replica set instance. If you connect to this node, you can perform only read operations on the databases of the replica set instance. • Connection String URI: ApsaraDB for MongoDB allows you to use a connection string URI to connect to a replica set instance to achieve load balancing and high availability.
Connection string	The connection string of a primary or secondary node is in the following format: `<host>:<port>` • <host>: the endpoint used to connect to the instance. • <port>: the port used to connect to the instance.
Connection string URI	A connection string URI is in the following format: `mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?options]]` • mongodb://: the prefix, indicating a connection string URI. • username:password@: the username and password used to log on to a database of the replica set instance. You must separate them with a colon (:). • hostX:portX: the endpoint and port of a node in the replica set instance. • /database: the database corresponding to the username and password if authentication is enabled. • ?options: additional connection options. **Note** If your application is in a production environment, we recommend that you use a connection string URI to connect to the instance. This way, when a node fails, the read and write operations of your application are not affected as a result of the failover.

Related information

- Connect to a replica set instance by using the mongo shell

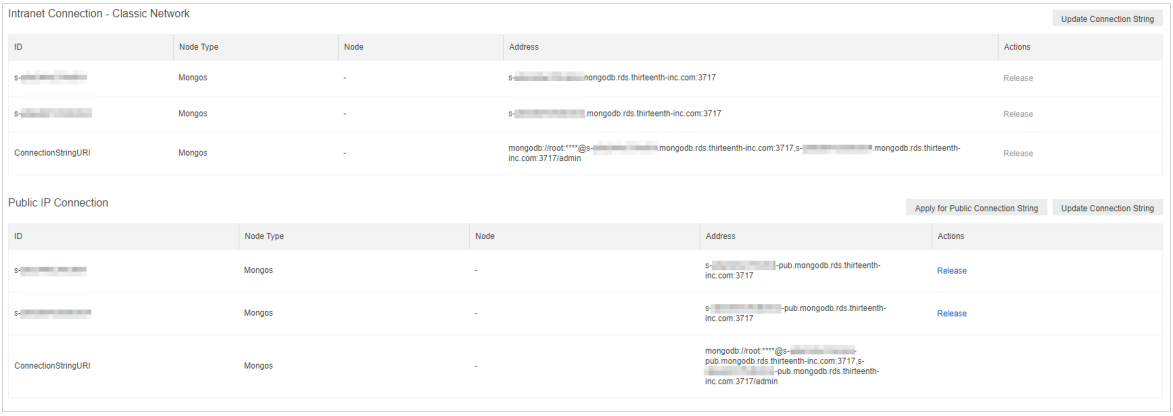
16.1.3.5. Overview of sharded cluster instance connections

ApsaraDB for MongoDB supports both connection strings and connection string URIs. You can use a connection string to connect to one mongos, and use a connection string URI to connect to more mongos. For high availability, we recommend that you use connection string URIs to connect your application to more mongos. This topic provides an overview of sharded cluster instance connections.

View connection strings

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.

3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, click **Database Connection** to view connection strings.



Description of connection strings

Item	Description
Type	• Intranet Connection - Classic Network: Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by security groups or whitelists. • Intranet Connection - VPC: A VPC is an isolated network with higher security and performance than a classic network. By default, ApsaraDB for MongoDB provides endpoints on a VPC. • Public IP Connection: Connecting to a sharded cluster instance over the Internet is risky. Therefore, ApsaraDB for MongoDB does not provide public endpoints. If you want to connect to an ApsaraDB for MongoDB instance from a device outside of Alibaba Cloud (such as a local device), you must apply for a public endpoint. For more information, see Apply for a public endpoint for an ApsaraDB for MongoDB instance.
Mongos ID	The connection string of a mongos is in the following format: `<host>:<port>` • <host>: the endpoint used to connect to the instance. • <port>: the port used to connect to the instance. **Note** During regular tests, you can use a connection string to directly connect to a mongos.

Item	Description
Connection string URI	A connection string URI is in the following format: `mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?options]]` • mongodb://: the prefix, indicating a connection string URI. • username:password@: the username and password used to log on to a database of the sharded cluster instance. You must separate the username and password with a colon (:). • hostX:portX: the endpoint and port of a mongos in the sharded cluster instance. • /database: the database corresponding to the username and password if authentication is enabled. • ?options: additional connection options. **Note** If your application is in a production environment, we recommend that you use a connection string URI to connect to the instance. Then your client can automatically distribute your requests to multiple mongos to balance loads. When a mongos fails, your client automatically redirects requests to other mongos in the normal state.

Log on to a database of an ApsaraDB for MongoDB instance

1. Obtain the connection string and the following information:
   - The username used to log on to the database. The initial username is root.
   - The password of the database user. If you forget the password of the root user, you can reset it. For more information, see Set a password for a sharded cluster instance.
   - The name of the database corresponding to the username if authentication is enabled. If the database username is root, enter admin.
2. Log on to the database.
   - Use DMS to log on to a replica set instance of ApsaraDB for MongoDB
   - Connect to a replica set instance by using the mongo shell

16.1.3.6. Use the mongo shell to connect to an ApsaraDB for MongoDB instance

This topic describes how to use the mongo shell to connect to an ApsaraDB for MongoDB instance.The mongo shell is a database management tool provided by ApsaraDB for MongoDB. You can install it on your client or in an ECS instance.

Prerequisites

- The version of the mongo shell is the same as your ApsaraDB for MongoDB instance. This ensures successful authentication. For information about the installation procedure, visit MongoDB official documentation. Choose the version in the upper-left corner of the page based on your client version.
- The IP address of your client is added to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.

3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, click **Database Connection** to view connection strings.

   > **Note**
   > - Replica set instances: Obtain the endpoint or connection string URI information of a node.
   > - Sharded cluster instances: Obtain the endpoint or connection string URI information of the mongos mode.
   >
   > For more information about connection strings, see Overview of replica set instance connections or Overview of sharded cluster instance connections.

5. Connect to the ApsaraDB for MongoDB instance from your client or ECS instance where the mongo shell is installed.

   Replica set instances:

   - Connection string of a node

     During regular tests, you can directly connect to a primary or secondary node. Take note that after the primary node fails, the system automatically switches to the secondary node, and the roles of connected nodes change. This affects the read and write operations of your application.

     Command format:

     ```
     mongo --host <host> -u <username> -p --authenticationDatabase <database>
     ```

     > **Note**
     > - <host>: the connection string of the primary or secondary node.
     > - <username>: the username you use to log on to a database of the instance. The initial username is **root**.
     > - <database>: the name of database corresponding to the username if authentication is enabled. If the database username is root, enter admin.

     Example:

     ```
     mongo --host dds-bp**********.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticationDatabase admin
     ```

     When `Enter password:` is displayed, enter the password of the database user and press Enter. If you forget the password of the root user, you can reset it. For more information, see Reset the password for an ApsaraDB for MongoDB instance.

     > **Note** The password characters are not displayed when you enter the password.

   - HA connection (recommended): You can use a connection string URI to connect to both the primary and secondary nodes of a replica set instance. This ensures that your application is always connected to the primary node and the read and write operations of your application are not affected even if the roles of the primary and secondary nodes are switched.

     Command format:

     ```
     mongo "<ConnectionStringURI>"
     ```

> **Note**
> - Double quotation marks ("") must be used.
> - <ConnectionStringURI>: the connection string URI of the instance.
>
> You must replace `****` in the connection string URI with the database password. For more information about how to set a database password, see Reset the password for an ApsaraDB for MongoDB instance.

Example:

```
mongo "mongodb://root:****@dds-**********.mongodb.rds.intra.env17e.shuguang.com:3717,dds-*****
*****.mongodb.rds.intra.env17e.shuguang.com:3717/admin? replicaSet=mgset-*****"
```

Sharded cluster instances:

- Connection string of the mongos node

  Command format:

  ```
  mongo --host <mongos_host> -u <username> -p --authenticationDatabase <database>
  ```

  > **Note**
  > - <mongos_host>: the connection string of a mongos node in the sharded cluster instance.
  > - <username>: the username you use to log on to a database of the instance. The initial username is **root**.
  > - <database>: the name of database corresponding to the username if authentication is enabled. If the database username is root, enter admin.

  Example:

  ```
  mongo --host s-bp**********.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticationDatabase
  admin
  ```

- HA connection (recommended): You can use a connection string URI to connect to a database. If one mongos node fails, another mongos node takes over business to ensure the high availability of the connection.

  Command format:

  ```
  mongo "<ConnectionStringURI>"
  ```

  > **Note**
  > - Double quotation marks ("") must be used.
  > - <ConnectionStringURI>: the connection string URI of the instance.
  >
  > You must replace `****` in the connection string URI with the database password. For more information about how to set a database password, see Reset the password for an ApsaraDB for MongoDB instance.

  Example:

  ```
  mongo "mongodb://root:****@s-**********.mongodb.rds.intra.env17e.shuguang.com:3717,s-*********
  *.mongodb.rds.intra.env17e.shuguang.com:3717/admin"
  ```

16.1.4. Instances

16.1.4.1. Create an ApsaraDB for MongoDB instance

This topic describes how to create an instance in the ApsaraDB for MongoDB console.

Prerequisites

An account is obtained to log on to the ApsaraDB for MongoDB console.

Create a replica set instance

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. On the **Replica Set Instances** page, click **Create Instance** in the upper-left corner. On the **Create ApsaraDB for MongoDB Instance** page, configure the parameters.

   The following table describes the required parameters.

   Parameters for creating a replica set instance

Section	Parameter	Description
Basic Settings	Organization	Select an organization for the new instance.
	Resource Set	Select a resource set for the new instance.
Region	Region	Select a region for the new instance.
	Zone	Select a zone for the new instance
Specifications	Database Engine	Select a database engine for the new instance. In this case, you can select only **MongoDB**.
	Engine Version	Select a database engine version for the new instance. Valid values: ○ 3.0 ○ 3.4 ○ 4.0 ○ 4.2
	Node Type	ApsaraDB for MongoDB supports the following options: ○ Three-node Replica Set: uses dedicated memory and I/O resources but shares CPU and storage resources with other general-purpose instances on the same server. ○ Dedicated Instance: uses dedicated CPU, memory, storage, and I/O resources to ensure long-term performance. In this case, an instance is not affected by other instances on the same server. ○ Dedicated Host: exclusively uses all resources of a server. This is the top configuration of exclusive specifications.
	Node Specifications	Select a node specification for the new instance. For more information, see descriptions in the ApsaraDB for MongoDB console.

Section	Parameter	Description
	Storage Capacity (GB)	The storage space of the instance, which contains the space for data, system files, log files, and transaction files. For more information, see *Instance types* in *ApsaraDB for MongoDB Product Introduction*.
Network	**Network Type**	ApsaraDB for MongoDB supports the following options: ○ **Classic Network**: Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by security groups or whitelists. ○ **VPC**: Virtual Private Cloud (VPC) is an isolated network environment built in Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for improved security. **Note** If you select the **VPC** network type, you must configure the **VPC** and **vSwitch** parameters.
	VPC	Select a VPC. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.
	vSwitch	Select a vSwitch. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.
Password Settings	**Instance Name**	Set the name of the new instance. The name must be 2 to 256 characters in length, The name must start with a letter, and can contain digits, letters, underscores (_), and hyphens (-).
	Password Setting	Determine when to set the password for logging on to databases in the new instance. You can select **Set Now** to set the logon password immediately, or select **Set after Purchase** to set the logon password after you create the instance. For more information, see Reset the password for an ApsaraDB for MongoDB instance.
	Logon Password	Set a password. The password must meet the following requirements: ○ It must be 8 to 32 characters in length. ○ It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ The name can contain special characters. Special characters include ! # $ % ^ & * ( ) _ + =
	Confirm Password	Enter the password again. The password you enter here must be the same as that in New Password.

3. Click **Submit** to create the instance.

Create a sharded cluster instance

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.

3. Click **Create Instance** in the upper-left corner to go to the **Create MongoDB Instance** page. Configure the parameters.

   The following table describes the required parameters.

   Parameters for creating a sharded cluster instance

Section	Parameter	MNS logs
Basic Settings	**Organization**	Select an organization for the new instance.
	Resource Set	Select a resource set for the new instance.
Region	**Region**	Select a region for the new instance.
	Zone	Select a zone for the new instance
Specifications	**Database Engine**	Select a database engine for the new instance. In this case, you can select only **MongoDB**.
	Engine Version	Select a database engine version for the new instance. Valid values: ○ 3.4 ○ 4.0 ○ 4.2
Network	**Network Type**	ApsaraDB for MongoDB supports the following options: ○ **Classic Network**: Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by security groups or whitelists. ○ **VPC**: Virtual Private Cloud (VPC) is an isolated network environment built in Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for improved security. **Note** If you select the **VPC** network type, you must configure the **VPC** and **vSwitch** parameters.
	VPC	Select a VPC. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.
	vSwitch	Select a vSwitch. If no vSwitch exists, you can click **Create vSwitch** to create a vSwitch. **Note** When **Network Type** is set to **VPC**, you must specify this parameter.
Mongos Specifications	**Mongos Specifications**	The specifications of the mongos node. For more information, see descriptions in the ApsaraDB for MongoDB console.
	Quantity	The number of the mongos nodes. You can select 2 to 32 mongos nodes.

Section	Parameter	MNS logs
Shard Specifications	Shard Specifications	The specifications of the shard node. For more information, see descriptions in the ApsaraDB for MongoDB console.
	Storage Capacity (GB)	The storage space of the instance, which contains the space for data, system files, log files, and transaction files. For more information, see *Instance types* in *ApsaraDB for MongoDB Product Introduction*.
	Quantity	The number of the shard nodes. You can select 2 to 32 shard nodes.
Config Server Specifications	Config Server Specifications	The specifications of Configserver nodes. It is fixed at **1 vCPU, 2 GiB memory** and cannot be customized.
	Storage Capacity (GB)	The storage space of the Configserver node. It is fixed at 20 GB and cannot be customized.
Password Settings	Instance Name	Set the name of the new instance. The name must be 2 to 256 characters in length, and can contain digits, letters, underscores (_), and hyphens (-). It must start with a letter.
	Password Setting	Determine when to set the password for logging on to databases in the new instance. You can select **Set Now** to set the logon password immediately, or select **Set after Purchase** to set the logon password after you create the instance. For more information, see Reset the password for an ApsaraDB for MongoDB instance.
	Logon Password	Set a password. The password must meet the following requirements: ○ It must be 8 to 32 characters in length. ○ It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ The name can contain special characters. Special characters include ! # $ % ^ & * ( ) _ + =
	Confirm Password	Enter the password again. The password you enter here must be the same as that in New Password.

4. Click **Submit** to create the instance.

16.1.4.2. View the details of an ApsaraDB for MongoDB instance

This topic describes how to view the details of an ApsaraDB for MongoDB instance, such as the basic information, internal network connection information, status, and configurations. This topic describes how to view the details of an ApsaraDB for MongoDB instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. Go to the instance details page. Either of the following methods can be used:

- Find the instance and click its ID to go to the **Basic Information** page, where you can view the details of the instance.
- In the Operations column corresponding to the instance, choose > **Manage** to go to the **Basic Information** page, where you can view the details of the instance.

16.1.4.3. Restart an ApsaraDB for MongoDB instance

You can manually restart an instance when the number of connections exceeds the threshold or any performance issue occurs on the instance. This topic describes how to restart an ApsaraDB for MongoDB instance.

Prerequisites

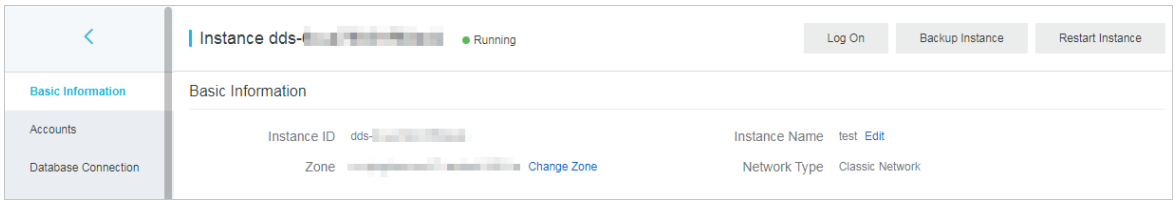
The instance is in the **Running** state.

Context

Note When an ApsaraDB for MongoDB instance is restarted, all its connections are terminated. Plan your operations in advance before you restart an ApsaraDB for MongoDB instance. Proceed with caution.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click in the **Actions** column corresponding to the instance and select **Manage**.
4. In the upper-right corner of the page, click **Restart Instance**.



Note You can also choose > **Restart** in the **Actions** column corresponding to the instance.

5. In the **Restart Instance** message, click **OK**.

16.1.4.4. Change the specifications of an ApsaraDB for MongoDB instance

This topic describes how to change the specifications of an ApsaraDB for MongoDB instance. You can upgrade or downgrade an ApsaraDB for MongoDB instance to meet your business needs.

Prerequisites

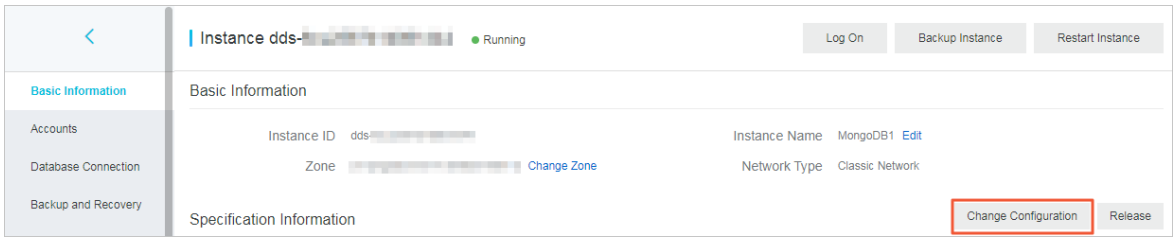
The instance must be an ApsaraDB for MongoDB replica set instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see

Log on to the ApsaraDB for MongoDB console.

2. On the **Replica Set Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
3. Click **Change Configuration** in the upper right corner of the Specification Information section to go to the **Modify Instance** page.



> **Note** To go to the **Modify Instance** page, you can also choose ⋮ > **Change Configuration** in the **Actions** column corresponding to the instance on the **Replica Set Instances** page.

4. On the **Modify Instance** page, change the instance specifications.

   You can change values of the following parameters:

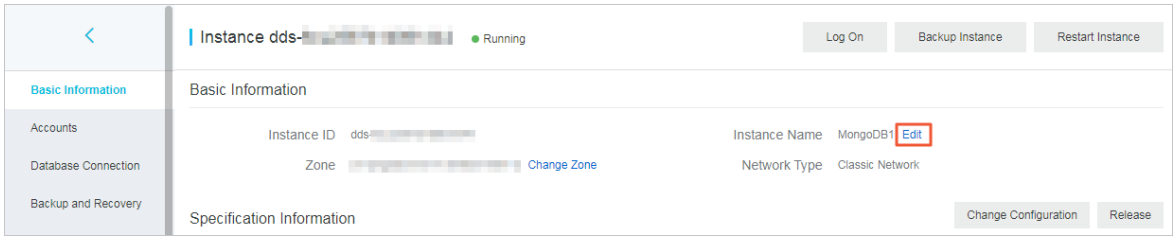
   - **Node Type**
   - **Node Specifications**
   - **Storage Capacity**
5. Click **Submit**.

16.1.4.5. Change the name of an ApsaraDB for MongoDB instance

This topic describes how to change the name of an ApsaraDB for MongoDB instance for better management.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. Click **Edit** next to **Instance Name**.



> **Note**
> - The instance name must start with an English letter. It cannot start with http:// or https://.
> - The instance name can contain letters, underscores (_), hyphens (-), and digits.
> - The instance name must be 2 to 128 characters in length.

5. Click **OK**.

16.1.4.6. Reset the password for an ApsaraDB for MongoDB instance

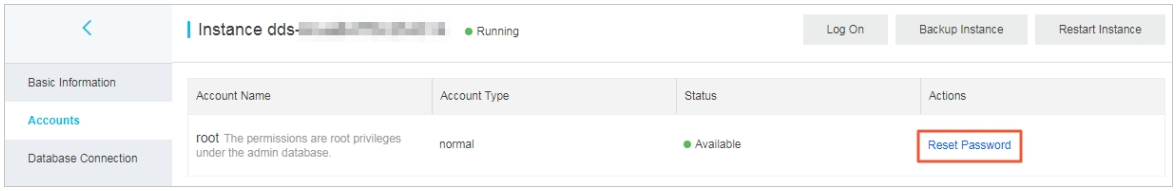
This topic describes how to reset your password in the ApsaraDB for MongoDB console.

Context

> **Notice** We recommend that you change your password on a regular basis to ensure data security.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Reset Password** in the Actions column and configure the parameters in the Reset Password panel.



Parameters for resetting a password describes the parameters.

Parameters for resetting a password

Parameter	Description
New Password	Specify the new password of the account based on the following rules: ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include !#$%^&*()_+= ○ The password must be 8 to 32 characters in length.
Confirm New Password	Enter the password again. The password you enter here must be the same as that in New Password.

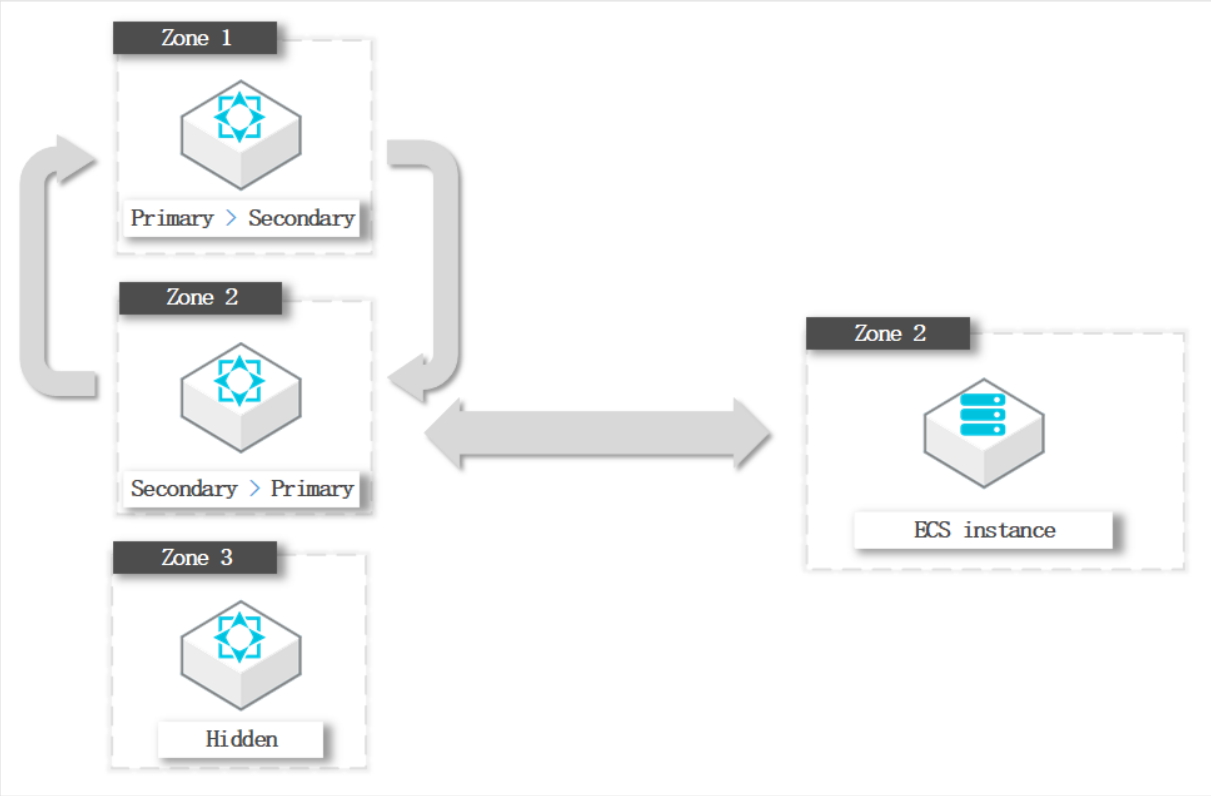
6. Click **OK**.

16.1.4.7. Switch node roles

You can switch the node roles of an ApsaraDB for MongoDB instance in the ApsaraDB for MongoDB console based on your business deployment.

Typical scenario

When an ECS instance and an ApsaraDB for MongoDB instance are in the same zone and connected over the internal network, the latency is minimal. If they are connected across different zones, the latency increases and the performance of ApsaraDB for MongoDB instances and your business will be affected.



In this example, the ECS instance to which the application belongs is in Zone 2. If the primary node of the ApsaraDB for MongoDB instance is in Zone 1, the ECS instance needs to connect to the primary node across zones.

To optimize the business deployment architecture, you can switch the roles of the primary and secondary nodes. In this example, you can change the role of the node in Zone 2 to primary and the role of the node in Zone 1 to secondary. Note that only the node roles are changed. ECS and ApsaraDB for MongoDB instances can be connected in the same zone without changing the actual zones and role IDs.

Precautions

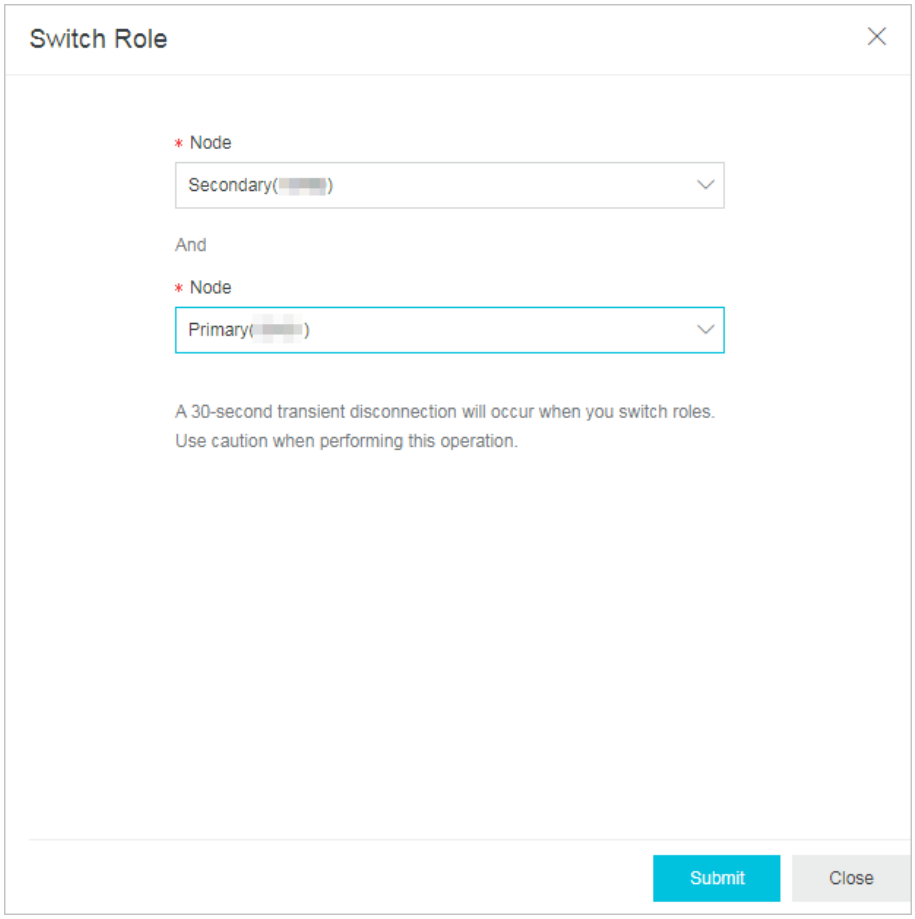
- Switching node roles will cause a transient disconnection of up to 30 seconds. Perform this operation during off-peak hours or ensure that your application has a reconnection mechanism.
- Switching node roles only changes the roles of nodes, but not the zones and role IDs of nodes.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in

the **Actions** column corresponding to the instance and select **Manage**.

4. In the left-side navigation pane, click **Service Availability**.
5. Subsequent steps on the **Service Availability** page vary depending on instance types.
   - Replica set instance
     a. Click **Switch Role** in the upper-right corner of the page.
     b. In the **Switch Role** dialog box that appears, select the nodes.

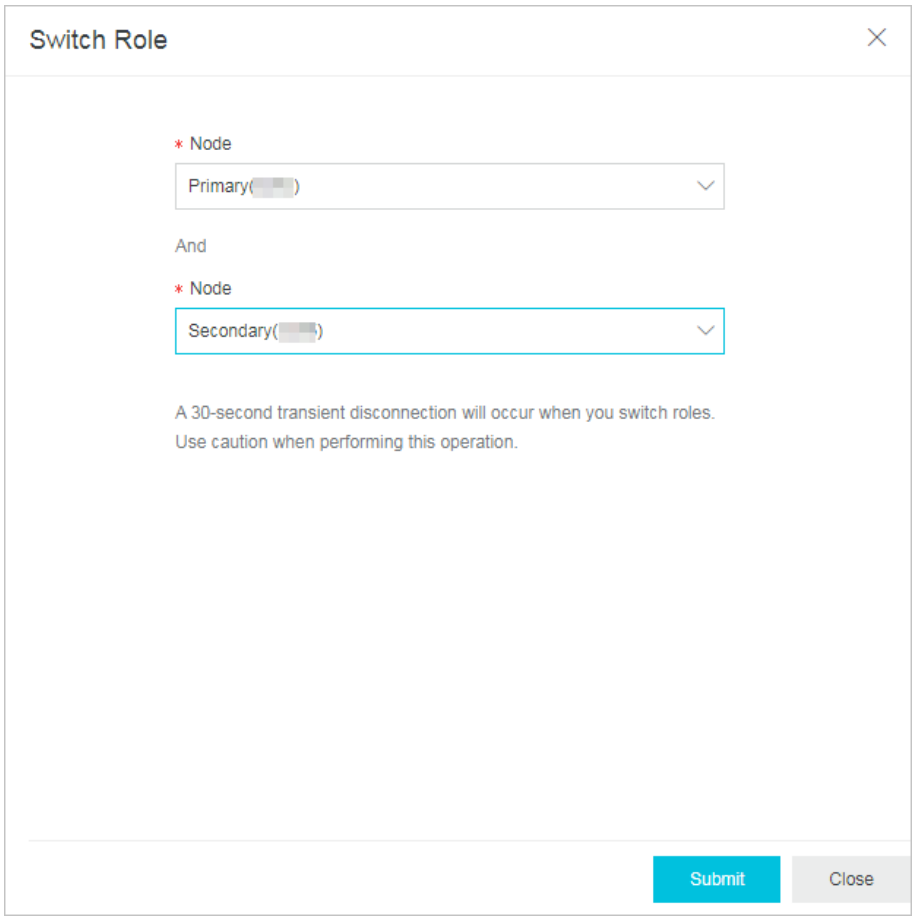


     c. Click **OK**.
   - Sharded cluster instance

> **Note** For sharded cluster instances, you can only manage the zone distribution of shard and Configserver nodes.

     a. In the upper-right corner of the **Zone Distribution for Shards** or **Zone Distribution for Configservers** section, click **Switch Role**.

b. In the **Switch Role** dialog box that appears, select the nodes.



- Click **OK**.

16.1.4.8. Release an ApsaraDB for MongoDB instance

This topic describes how to manually release an ApsaraDB for MongoDB instance to meet your business needs.
This topic describes how to manually release an ApsaraDB for MongoDB instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the lower-right corner of the **Basic Information** section, click **Release**.

   > **Note** You can also go to the Replica Set Instances page, find the instance, click the ⋮ icon in the **Actions** column, and then choose **Release**.

5. In the **Release Instance** message, click **OK**.

> **Warning** After you release an ApsaraDB for MongoDB instance, data in the instance can no longer be recovered. Proceed with caution.

16.1.4.9. Primary/secondary failover

16.1.4.9.1. Trigger a primary/secondary failover for a replica set instance

An ApsaraDB for MongoDB replica set instance consists of three nodes by default. ApsaraDB for MongoDB provides connection strings for you to connect to the primary node and a secondary node. The other secondary node is hidden as a backup to ensure high availability. If a node is faulty, the high availability system of ApsaraDB for MongoDB automatically triggers a primary/secondary failover to ensue the availability of the instance. You also can manually trigger a primary/secondary failover for an ApsaraDB for MongoDB instance in scenarios such as routine disaster recovery drills.

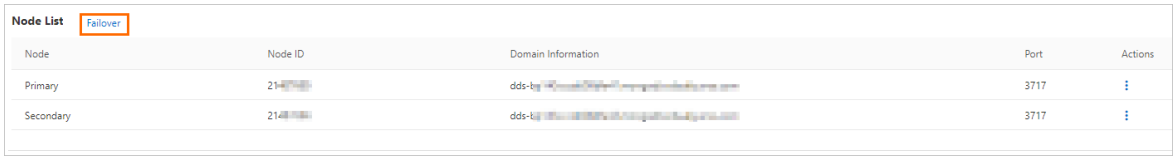
Context

After you log on to the ApsaraDB for MongoDB console or call the SwitchDBInstanceHA operation to trigger a primary/secondary failover for a replica set instance, ApsaraDB for MongoDB interchanges the roles of the primary and secondary nodes.

> **Note**
> - You can trigger a primary/secondary failover only for replica set and sharded cluster instances, but not for standalone instances due to their single-node architecture.
> - Each time you trigger a primary/secondary failover for an instance, the instance may have a transient connection error of about 30 seconds. Ensure that your applications can automatically re-establish a connection.
> - You can trigger a primary/secondary failover only for instances in the running state.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. On the **Replica Set Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
3. In the **Node List** section, click **Failover**, as shown in the following figure.



4. In the dialog box that appears, click **OK**.
5. The instance status changes to **HA Switching**.The failover is successful when the instance status changes back to **Running**.

   The failover operation is complete in about one minute. Then the instance returns to normal.

> **Note** If you have connected to the connection string of the primary node for an instance, you are connecting to a secondary node after a failover and you have no write permissions on the instance. In this case, you must connect to the connection string of the new primary node and obtain read and write permissions. For more information, see Overview of replica set instance connections.

16.1.4.9.2. Trigger a primary/secondary failover for a sharded cluster instance

Each shard or Configserver of a sharded cluster instance consists of three nodes by default. If a node is faulty, the high availability system of ApsaraDB for MongoDB automatically triggers a primary/secondary failover to ensure the availability of the shard. You can also manually trigger a primary/secondary failover for an ApsaraDB for MongoDB instance in scenarios such as routine disaster recovery drills.

Precautions

ApsaraDB for MongoDB provides connection strings for you to connect to the primary and secondary nodes. The hidden node is invisible to you and only used to ensure high availability. After you log on to the ApsaraDB for MongoDB console or call the SwitchDBInstanceHA operation to trigger a primary/secondary failover for a shard of a sharded cluster instance, ApsaraDB for MongoDB switches the roles of the primary and secondary nodes.

> **Note**
> - You can trigger a primary/secondary failover only for the shard or Configserve node in the running state.
> - Each time you trigger a primary/secondary failover for an instance, the instance may have a transient connection error of about 30 seconds. We recommend that you perform this operation during off-peak hours and ensure that your applications can automatically re-establish a connection.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the **Shard List** or **ConfigServer List** section, find the node and choose ⋮ > **Failover** in the Actions column.

   > **Note** You can trigger a primary/secondary failover separately for each shard node. The failover operation takes effect only for the current shard node and does not affect other shard nodes of the same sharded cluster instance.

5. In the **Failover** message, click **OK**.

   > **Note** The failover operation is complete in about one minute.

16.1.4.10. Monitoring

This topic describes the performance metrics provided by ApsaraDB for MongoDB to check the status of ApsaraDB for MongoDB instances. You can view these performance metrics in the ApsaraDB for MongoDB console.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click  in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, click **Monitoring Info**.

   You can select a time range to query historical performance metrics. The following table describes metric details.

   Performance metrics

Performance metric	Description	Monitoring frequency
CPU Utilization Percentage	cpu_usage: the CPU utilization of the instance	Every 300 seconds
Memory Usage Percentage	mem_usage: the memory usage of the instance	Every 300 seconds
IOPS Usage	The input/output operations per second (IOPS) of the instance. The following items are included: ◦ data_iops: the IOPS of the data disk. ◦ log_iops: the IOPS of the log disk.	Every 300 seconds
IOPS Usage Percentage	iops_usage: the ratio of the IOPS used by the instance to the maximum IOPS allowed	Every 300 seconds
Disk Usage	The total disk space used by the instance. The following items are included: ◦ ins_size: the total space used. ◦ data_size: the space used on the data disk. ◦ log_size: the space used on the log disk.	Every 300 seconds
Disk Usage Percentage	disk_usage: the ratio of the total disk space used by the instance to the maximum disk space that can be used	Every 300 seconds

Performance metric	Description	Monitoring frequency
Operation QPS	The queries per second (QPS) of the instance. The following items are included: ◦ The number of insert operations. ◦ The number of query operations. ◦ The number of delete operations. ◦ The number of update operations. ◦ The number of getmore operations. ◦ The number of command operations.	Every 300 seconds
Connections	current_conn: the number of current connections to the instance	Every 300 seconds
Cursors	The number of cursors used by the instance. The following items are included: ◦ total_open: the number of cursors that are opened. ◦ timed_out: the number of cursors that timed out.	Every 300 seconds
Network Traffic	The network traffic of the instance. The following items are included: ◦ bytes_in: the inbound network traffic. ◦ bytes_out: the outbound network traffic. ◦ num_requests: the number of requests that are processed.	Every 300 seconds
Global Lock Waiting Queues	The length of the queues that are waiting for global locks for the instance. The following items are included: ◦ gl_cq_total: the length of the queue that is waiting for both global read and write locks. ◦ gl_cq_readers: the length of the queue that is waiting for global read locks. ◦ gl_cq_writers: the length of the queue that is waiting for global write locks.	Every 300 seconds

Performance metric	Description	Monitoring frequency
WiredTiger	The cache metrics of the WiredTiger engine used by the instance. The following items are included: ○ bytes_read_into_cache: the amount of data that is read into the cache. ○ bytes_written_from_cache: the amount of data that is written from the cache to the disk. ○ maximum_bytes_configured: the size of the maximum available disk space that is configured.	Every 300 seconds
Primary/Secondary Replication Latency	repl_lag: the latency in data synchronization between the primary and secondary nodes of the instance	Every 300 seconds

16.1.5. Backup and restoration

16.1.5.1. Configure automatic backup for an ApsaraDB for MongoDB instance

This topic describes how to configure automatic backup for an ApsaraDB for MongoDB instance. ApsaraDB for MongoDB automatically backs up data based on the backup policy you specify.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, click **Backup and Recovery**.
5. In the upper-left corner of the page, click **Backup Settings**. Configure the parameters in the Backup Settings panel.

   The following table describes the parameters.

   Backup policy parameters

Parameter	Description
Retention Days	The number of days for which you want to retain backup files. It can only be seven days.
Backup Time	The hour at which you want to perform the backup task.
Day of Week	The backup cycle. You can select one or more days in a week.

6. Click **OK**.

16.1.5.2. Manually back up an ApsaraDB for MongoDB instance

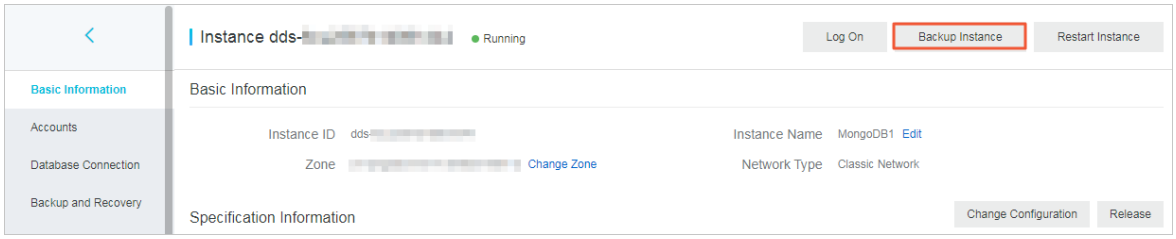
This topic describes how to manually back up an ApsaraDB for MongoDB instance.

Backup methods

- Physical backup: This method backs up physical database files of an ApsaraDB for MongoDB instance. Compared with logical backup, physical backup provides faster data backup and recovery.
- Logical backup: The mongodump tool is used to store operation records of databases to a logical backup file. Then, data can be restore data by using the mongorestore tool.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the upper-right corner of the page, click **Backup Instance**.



5. In the dialog box that appears, set **Backup Method** and click **OK**.

16.1.5.3. Restore data to the current ApsaraDB for MongoDB instance

This topic describes how to restore data to the current ApsaraDB for MongoDB instance. This helps minimize the data loss caused by incorrect operations.

Prerequisites

The instance is a replica set instance with three nodes.

Background information

- The time required to restore data to your current instance varies depending on factors such as the data volume, task queue status, and network conditions. When the status of the instance changes to **Running**, the restoration is complete.
- If you restore data to your current instance, all existing data is overwritten and cannot be restored.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. On the **Replica Set Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column

corresponding to the instance and select **Manage**.

3. In the left-side navigation pane, click **Backup and Recovery**.
4. On the **Backup and Recovery** page that appears, find the backup set and choose > **Data Recovery** in the Actions column.

   > **Note** If you have upgraded the database version, you cannot use the backup files of the earlier database version to restore data.

5. In the **Roll Back Instance** message, click **OK**.

   > **Note** The instance status becomes **Restoring from Backup** after you click **OK**. You can click **Refresh** in the upper-right corner of the **Backup and Recovery** page to update the instance status. The restoration is complete when the instance status changes to **Running**.

16.1.6. Database connection

16.1.6.1. Modify a public or internal endpoint of an ApsaraDB for MongoDB instance

This topic describes how to modify the public and internal endpoints of an ApsaraDB for MongoDB instance in the ApsaraDB for MongoDB console.

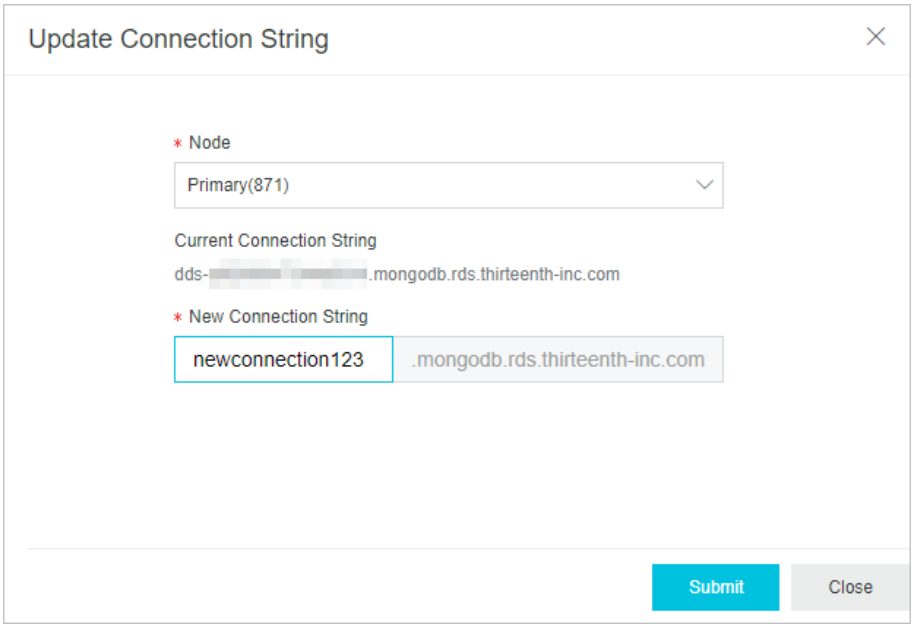
Limits

Architecture	Limit
Replica set instance	You can modify the public and internal endpoints of both primary and secondary nodes.
Sharded cluster instance	You can only modify the public and internal endpoints of a mongos.

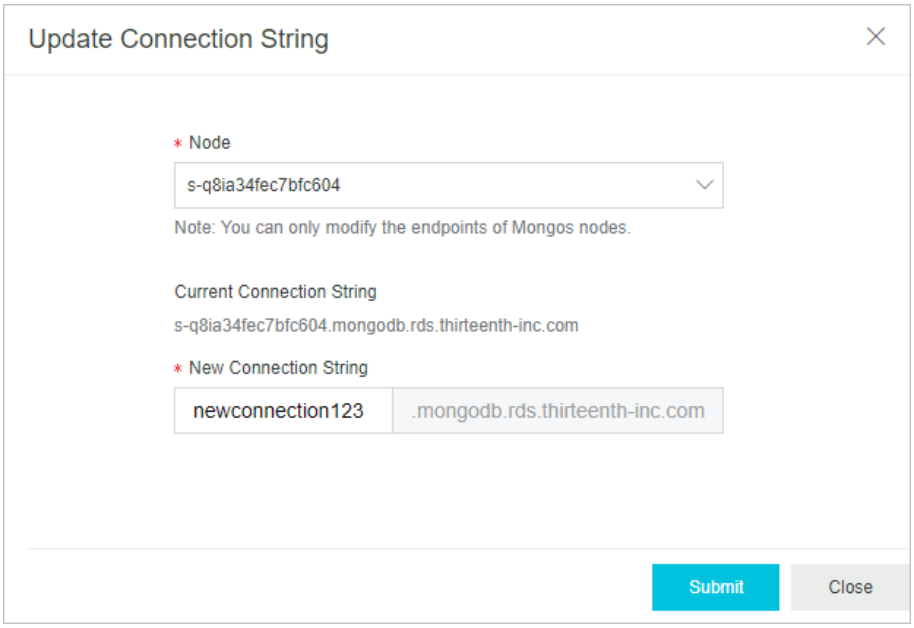
Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, click **Database Connection**.
5. In the **Intranet Connection** or **Public IP Connection** section, click **Update Connection String**.
6. In the dialog box that appears, enter a new endpoint.

   Replica set instance



Sharded cluster instance



> **Note**
> - You can only modify the prefix of the endpoint.
> - It must start with a lowercase letter and end with a letter or a digit. The password must be 8 to 64 characters in length and can contain lowercase letters, digits, and hyphens (-).

7. Click **Submit**.

What's next

After you modify the public or internal endpoint, you must connect a client or an application to your ApsaraDB for MongoDB instance by using the new endpoint.

16.1.6.2. Use the mongo shell to connect to an ApsaraDB for MongoDB instance

This topic describes how to use the mongo shell to connect to an ApsaraDB for MongoDB instance.The mongo shell is a database management tool provided by ApsaraDB for MongoDB. You can install it on your client or in an ECS instance.

Prerequisites

- The version of the mongo shell is the same as your ApsaraDB for MongoDB instance. This ensures successful authentication. For information about the installation procedure, visit MongoDB official documentation. Choose the version in the upper-left corner of the page based on your client version.
- The IP address of your client is added to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, click **Database Connection** to view connection strings.

   > **Note**
   > - Replica set instances: Obtain the endpoint or connection string URI information of a node.
   > - Sharded cluster instances: Obtain the endpoint or connection string URI information of the mongos mode.
   >
   > For more information about connection strings, see Overview of replica set instance connections or Overview of sharded cluster instance connections.

5. Connect to the ApsaraDB for MongoDB instance from your client or ECS instance where the mongo shell is installed.

   Replica set instances:

   - Connection string of a node

     During regular tests, you can directly connect to a primary or secondary node. Take note that after the primary node fails, the system automatically switches to the secondary node, and the roles of connected nodes change. This affects the read and write operations of your application.

     Command format:

     ```
     mongo --host <host> -u <username> -p --authenticationDatabase <database>
     ```

> **Note**
> - <host>: the connection string of the primary or secondary node.
> - <username>: the username you use to log on to a database of the instance. The initial username is **root**.
> - <database>: the name of database corresponding to the username if authentication is enabled. If the database username is root, enter admin.

Example:

```
mongo --host dds-bp**********.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticationDatabase admin
```

When `Enter password:` is displayed, enter the password of the database user and press Enter. If you forget the password of the root user, you can reset it. For more information, see Reset the password for an ApsaraDB for MongoDB instance.

> **Note** The password characters are not displayed when you enter the password.

- HA connection (recommended): You can use a connection string URI to connect to both the primary and secondary nodes of a replica set instance. This ensures that your application is always connected to the primary node and the read and write operations of your application are not affected even if the roles of the primary and secondary nodes are switched.

  Command format:

  ```
  mongo "<ConnectionStringURI>"
  ```

  > **Note**
  > - Double quotation marks ("") must be used.
  > - <ConnectionStringURI>: the connection string URI of the instance.
  >
  >   You must replace `****` in the connection string URI with the database password. For more information about how to set a database password, see Reset the password for an ApsaraDB for MongoDB instance.

  Example:

  ```
  mongo "mongodb://root:****@dds-**********.mongodb.rds.intra.env17e.shuguang.com:3717,dds-**********.mongodb.rds.intra.env17e.shuguang.com:3717/admin? replicaSet=mgset-*****"
  ```

Sharded cluster instances:

- Connection string of the mongos node

  Command format:

  ```
  mongo --host <mongos_host> -u <username> -p --authenticationDatabase <database>
  ```

> **Note**
> - <mongos_host>: the connection string of a mongos node in the sharded cluster instance.
> - <username>: the username you use to log on to a database of the instance. The initial username is **root**.
> - <database>: the name of database corresponding to the username if authentication is enabled. If the database username is root, enter admin.

Example:

```
mongo --host s-bp**********.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticationDatabase admin
```

- HA connection (recommended): You can use a connection string URI to connect to a database. If one mongos node fails, another mongos node takes over business to ensure the high availability of the connection.

  Command format:

  ```
  mongo "<ConnectionStringURI>"
  ```

  > **Note**
  > - Double quotation marks ("") must be used.
  > - <ConnectionStringURI>: the connection string URI of the instance.
  >
  >   You must replace `****` in the connection string URI with the database password. For more information about how to set a database password, see Reset the password for an ApsaraDB for MongoDB instance.

  Example:

  ```
  mongo "mongodb://root:****@s-**********.mongodb.rds.intra.env17e.shuguang.com:3717,s-*********.mongodb.rds.intra.env17e.shuguang.com:3717/admin"
  ```

16.1.6.3. Use DMS to log on to an ApsaraDB for MongoDB instance

You can use DMS to connect to an ApsaraDB for MongoDB instance.

Prerequisites

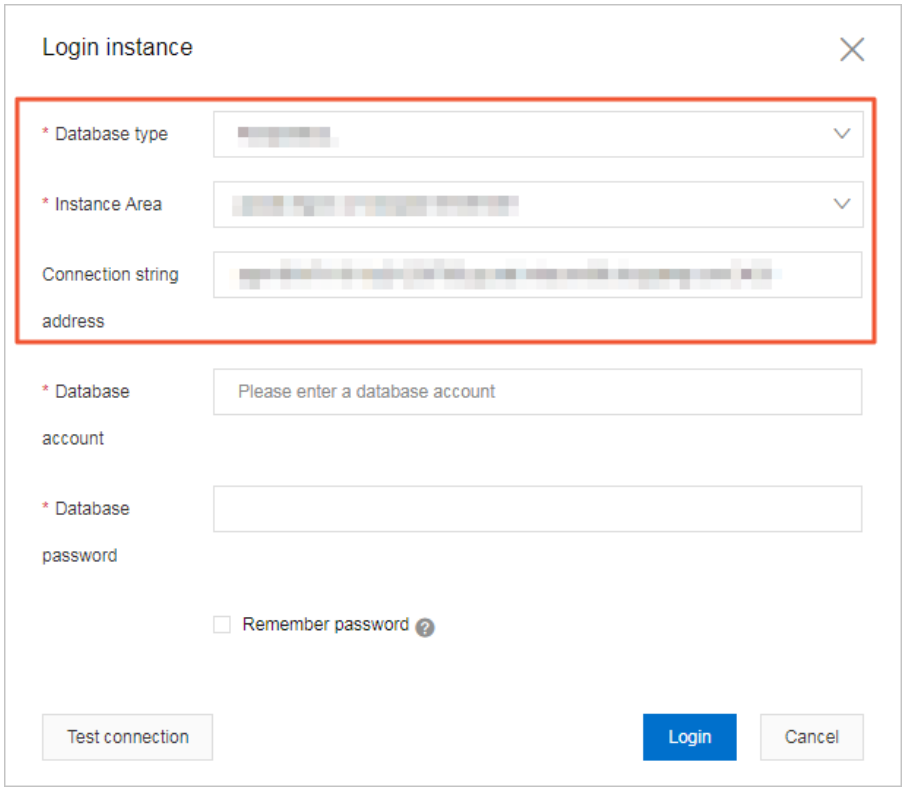
The IP address whitelist is configured. For more information about how to configure the IP address whitelist, see Configure a whitelist for an ApsaraDB for MongoDB instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. Click **Log On** in the upper-right corner of the page.

> **Note** For a sharded cluster instance, you must also select the Mongos node.

5. In the **Login instance** dialog box of the **DMS** console, check values of **Database type**, **Instance Area**, and **Connection string address**. If the information is correct, enter **Database account** and **Database password**, as shown in the following figure.



Parameter	Description
Database type	The engine of the database. By default, the engine of the database to be connected is displayed.
Instance Area	The region where the instance is deployed. By default, the region of the current instance is displayed.
Connection string address	The endpoint of the instance. By default, the endpoint of the current instance is displayed.
Database account	The account of the database to be connected.
Database password	The password of the account used to connect to the database.

6. Click **Login**.

> **Note** If you want your web browser to remember the password, select **Remember password** before you click **Login**.

16.1.6.4. Apply for a public endpoint for an ApsaraDB for MongoDB instance

This topic describes how to apply for a public endpoint for an ApsaraDB for MongoDB instance when you want to connect to this instance over the Internet.

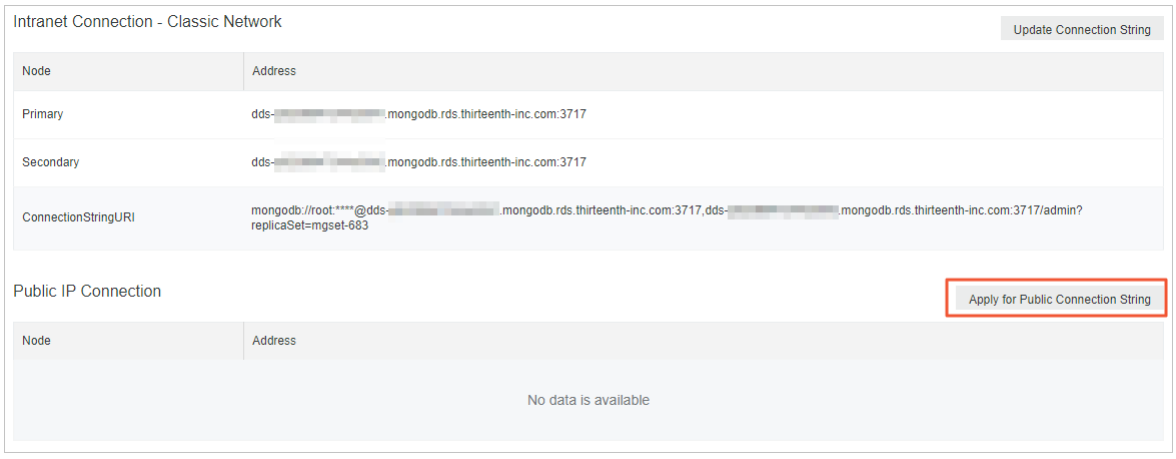
Context

The following table describes the VPC and classic network endpoints supported by ApsaraDB for MongoDB.

Type	Description
VPC endpoint	• A VPC is an isolated network with higher security and performance than a classic network. • By default, ApsaraDB for MongoDB provides endpoints on a VPC.
Classic network endpoint	Cloud services on a classic network are not isolated. Unauthorized access can only be blocked by using security groups or whitelists.
Public endpoint	• It is risky to connect to an ApsaraDB for MongoDB instance over the Internet. Therefore, ApsaraDB for MongoDB does not provide public endpoints. • If you want to connect to a replica set instance from a device outside Alibaba Cloud (for example, a local client), you must apply for a public endpoint.

Apply for a public endpoint for a replica set instance

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. On the **Replica Set Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
3. In the left-side navigation pane, click **Database Connection**.
4. In the **Public IP Connection** section, click **Apply for Public Connection String**.



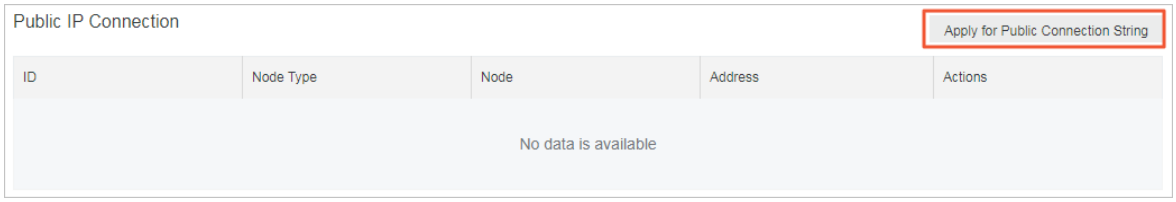
5. In the **Apply for Public Connection String** message that appears, click **OK**.

> **Note** If you want to connect to an ApsaraDB for MongoDB (Serverless) instance by using a public endpoint, you must add the public IP address of your client to a whitelist of this instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

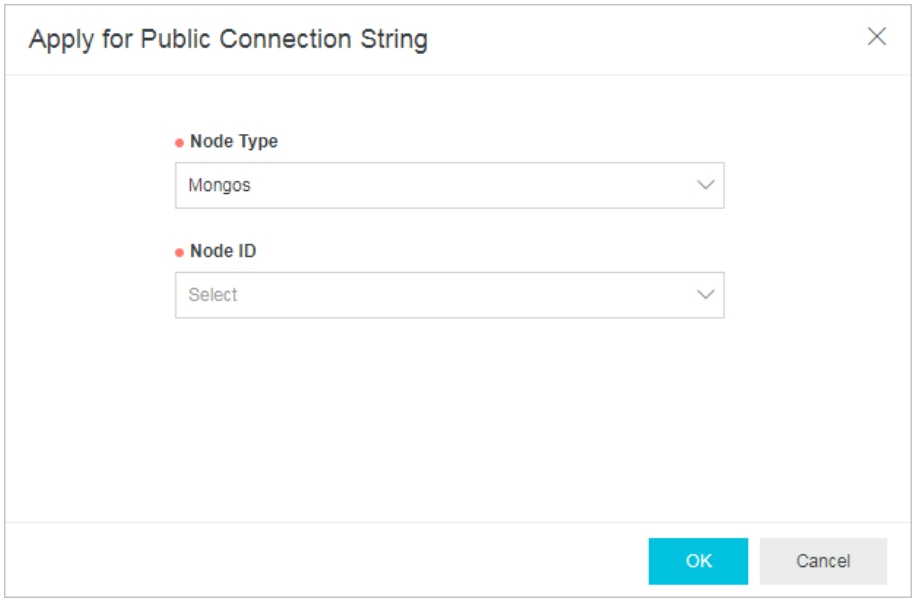
After the application is complete, the replica set instance generates new endpoints for both the primary and secondary nodes and the corresponding connection string URI. For more information, see Overview of replica set instance connections.

Apply for a public endpoint for a sharded cluster instance

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, click **Database Connection**.
5. In the **Public IP Connection** section, click **Apply for Public Connection String**.



6. In the dialog box that appears, specify **Node Type** and **Node ID**, and click **OK**.



Parameter	Value	Description
Mongos	The mongos node. You can only apply for public endpoints for mongos nodes. Your applications are connected to mongos nodes in most cases.	
Node ID	The ID of the component for which you want to apply for a public endpoint.	None

Note To apply for a public endpoint for other mongos, repeat this step. You can only apply for a new public endpoint after the current one is created.

References

- To ensure data security, we recommend that you release a public endpoint if you no longer need it. For more information, see Release a public connection string.
- Before you connect to a replica set instance over the Internet, we recommend that you enable SSL encryption. For more information, see Use the mongo shell to connect to an ApsaraDB for MongoDB database in SSL encryption mode.

16.1.6.5. Release a public connection string

To ensure data security, you can release a public connection string that is no longer needed in the console.

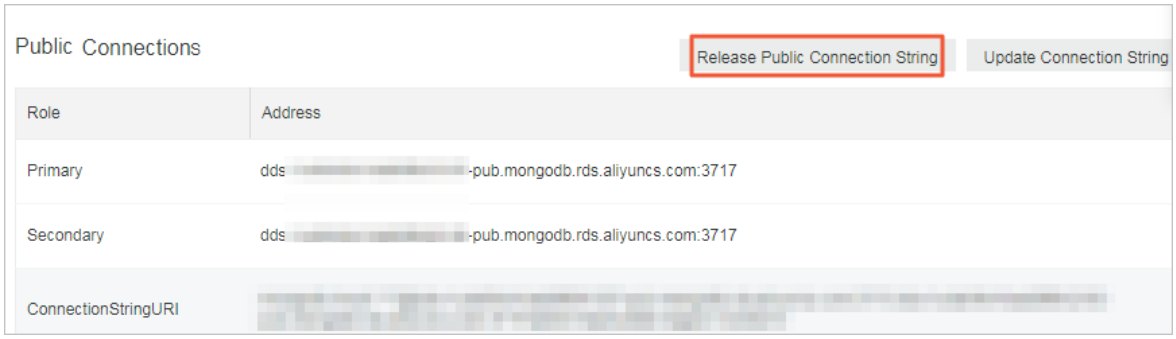
Precautions

- You can release one or more public connection strings of the mongos nodes for a sharded cluster instance.
- After the public connection string is released for an instance or node, you cannot connect to the instance or node through the original public connection string.
- After the public connection string is released, we recommend that you delete the corresponding public IP address from the whitelist to ensure data security. For more information, see Configure a whitelist.

Release a public endpoint for a replica set instance

> **Note** After the public connection string of a replica set instance is released, the public connection strings of the primary and secondary nodes are released.

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. On the **Replica Set Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
3. On the page that appears, click **Database Connection** in the left-side navigation pane.
4. In the **Public IP Connection** section, click **Release Public Connection String**.



5. In the message that appears, click **OK**.

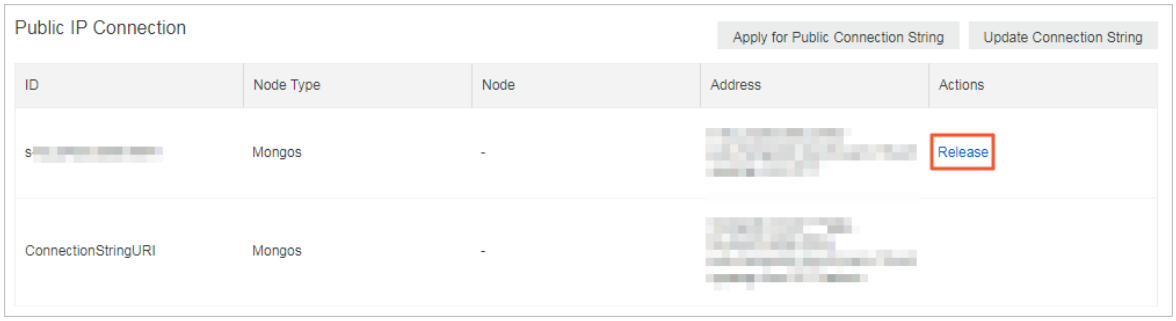
Release a public endpoint for a sharded cluster instance

You can release one or more public connection strings of the mongos, shard, and Configserver nodes for a sharded cluster instance.

> **Note** After the public connection string of a shard or Configserver node is released, the public connection strings of the primary and secondary nodes are released.

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.

2. In the left-side navigation pane, click **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, click **Database Connection**.
5. In the **Public IP Connection** section, find the mongos, shard, or Configserver node for which you want to release the public connection string.
6. In the **Actions** column corresponding to the instance, click **Release**.



> **Note** You can repeat this step to release the public connection strings of other nodes. To release the public connection string of the next node, you must wait until the public connection string of the current node is released or the status of the current node becomes **Running**.

7. In the message that appears, click **OK**.

16.1.6.6. Overview of replica set instance connections

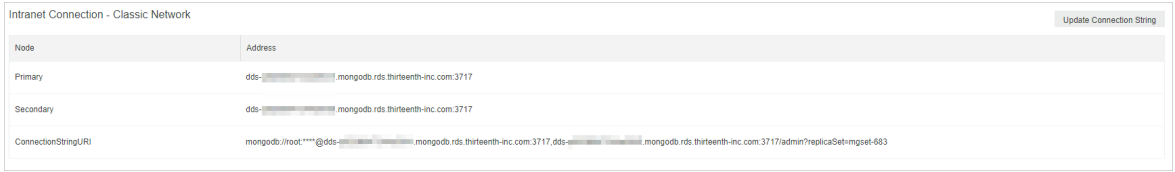
ApsaraDB for MongoDB supports both connection strings and connection string URIs. You can use a connection string to connect to either the primary or secondary node, and use a connection string URI to connect to both of them. For high availability, we recommend that you use connection string URIs to connect your application to both primary and secondary nodes. This topic provides an overview of replica set instance connections.

Prerequisites

A whitelist is configured for the instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

View connection strings

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. On the **Replica Set Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
3. In the left-side navigation pane, click **Database Connection** to view connection strings.



Description of connection strings

Item	Description
Type	• Intranet Connection - Classic Network: Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by security groups or whitelists. • Intranet Connection - VPC: A VPC is an isolated network with higher security and performance than a classic network. By default, ApsaraDB for MongoDB provides endpoints on a VPC.
Role	• Primary: the primary node in the replica set instance. If you connect to this node, you can perform read and write operations on the databases of the replica set instance. • Secondary: the secondary node in the replica set instance. If you connect to this node, you can perform only read operations on the databases of the replica set instance. • Connection String URI: ApsaraDB for MongoDB allows you to use a connection string URI to connect to a replica set instance to achieve load balancing and high availability.
Connection string	The connection string of a primary or secondary node is in the following format: `<host>:<port>` • <host>: the endpoint used to connect to the instance. • <port>: the port used to connect to the instance.
Connection string URI	A connection string URI is in the following format: `mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?options]]` • mongodb://: the prefix, indicating a connection string URI. • username:password@: the username and password used to log on to a database of the replica set instance. You must separate them with a colon (:). • hostX:portX: the endpoint and port of a node in the replica set instance. • /database: the database corresponding to the username and password if authentication is enabled. • ?options: additional connection options. **Note** If your application is in a production environment, we recommend that you use a connection string URI to connect to the instance. This way, when a node fails, the read and write operations of your application are not affected as a result of the failover.

Related information

- Connect to a replica set instance by using the mongo shell

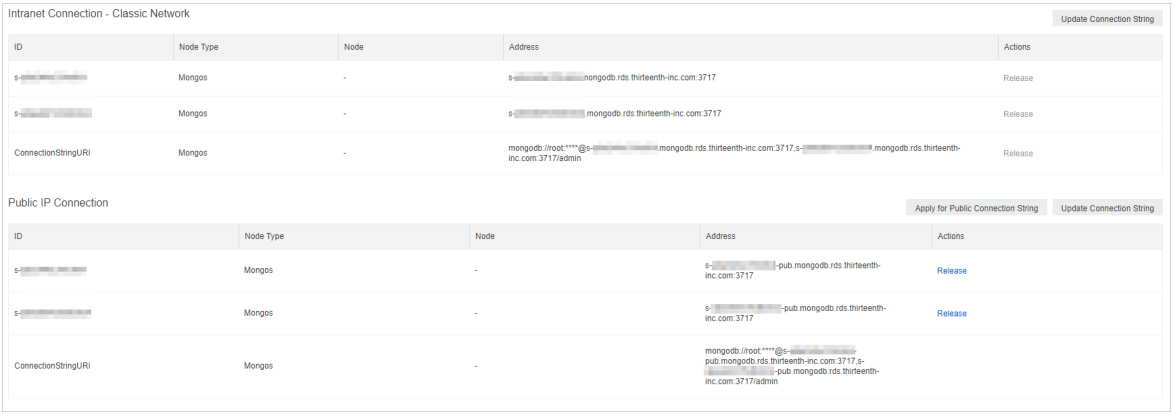
16.1.6.7. Overview of sharded cluster instance connections

ApsaraDB for MongoDB supports both connection strings and connection string URIs. You can use a connection string to connect to one mongos, and use a connection string URI to connect to more mongos. For high availability, we recommend that you use connection string URIs to connect your application to more mongos. This topic provides an overview of sharded cluster instance connections.

View connection strings

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.

3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, click **Database Connection** to view connection strings.



Description of connection strings

Item	Description
Type	• Intranet Connection - Classic Network: Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by security groups or whitelists. • Intranet Connection - VPC: A VPC is an isolated network with higher security and performance than a classic network. By default, ApsaraDB for MongoDB provides endpoints on a VPC. • Public IP Connection: Connecting to a sharded cluster instance over the Internet is risky. Therefore, ApsaraDB for MongoDB does not provide public endpoints. If you want to connect to an ApsaraDB for MongoDB instance from a device outside of Alibaba Cloud (such as a local device), you must apply for a public endpoint. For more information, see Apply for a public endpoint for an ApsaraDB for MongoDB instance.
Mongos ID	The connection string of a mongos is in the following format: `<host>:<port>` • <host>: the endpoint used to connect to the instance. • <port>: the port used to connect to the instance. **Note** During regular tests, you can use a connection string to directly connect to a mongos.

Item	Description
Connection string URI	A connection string URI is in the following format: `mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?options]]` • mongodb://: the prefix, indicating a connection string URI. • username:password@: the username and password used to log on to a database of the sharded cluster instance. You must separate the username and password with a colon (:). • hostX:portX: the endpoint and port of a mongos in the sharded cluster instance. • /database: the database corresponding to the username and password if authentication is enabled. • ?options: additional connection options. **Note** If your application is in a production environment, we recommend that you use a connection string URI to connect to the instance. Then your client can automatically distribute your requests to multiple mongos to balance loads. When a mongos fails, your client automatically redirects requests to other mongos in the normal state.

Log on to a database of an ApsaraDB for MongoDB instance

1. Obtain the connection string and the following information:
   - The username used to log on to the database. The initial username is root.
   - The password of the database user. If you forget the password of the root user, you can reset it. For more information, see Set a password for a sharded cluster instance.
   - The name of the database corresponding to the username if authentication is enabled. If the database username is root, enter admin.
2. Log on to the database.
   - Use DMS to log on to a replica set instance of ApsaraDB for MongoDB
   - Connect to a replica set instance by using the mongo shell

16.1.7. Data security

16.1.7.1. Configure a whitelist for an ApsaraDB for MongoDB instance

This topic describes how to configure a whitelist for an ApsaraDB for MongoDB instance. Before you use an ApsaraDB for MongoDB instance, you must add the IP addresses or Classless Inter-Domain Routing (CIDR) blocks that you use for database access to a whitelist of this instance. This improves database security and stability. Proper configuration of whitelists can enhance access security of ApsaraDB for MongoDB. We recommend that you maintain the whitelists on a regular basis.

Context

The system creates a default whitelist for each instance. This whitelist can be modified or cleared, but it cannot be deleted. After an ApsaraDB for MongoDB instance is created, the system automatically adds the IP address 127.0.0.1 to the **default** whitelist of this instance. The IP address 0.0.0.0/0 indicates that all IP addresses are allowed to access this instance. You must add the IP addresses or CIDR blocks that you allow to access this ApsaraDB for MongoDB instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, choose **Data Security > Whitelist Settings**.
5. You can manually configure a whitelist or import ECS Internal IP addresses to the whitelist.

   Manually modify a whitelist

   i. Find the whitelist you want to modify and choose **> Manually Modify** in the **Actions** column.

   ii. Enter IP addresses or CIDR blocks.

   > Note
   > - Separate multiple IP addresses with commas (,). You can add a maximum of 1,000 different IP addresses to a whitelist. Supported formats are IP addresses such as 0.0.0.0/0 and 10.23.12.24, or CIDR blocks such as 10.23.12.24/24. /24 indicates the length of the IP address prefix in the CIDR block. An IP address prefix can contain 1 to 32 bits.
   > - If the IP whitelist is empty or only contains `0.0.0.0/0`, all devices are granted access. This is risky for your ApsaraDB for MongoDB instance. We recommend that you add only the IP addresses or CIDR blocks of your own web servers to the whitelist.

   iii. Click **OK**.

   Load IP addresses of ECS instances

   i. Find the whitelist, and choose **> Import ECS Intranet IP** in the **Actions** column.

   ii. From the displayed internal IP addresses of ECS instances that belong to the current account, select the IP addresses and click > to add them to the whitelist.

   iii. Click **OK**.

16.1.7.2. Add or delete a whitelist

This topic describes how to add or delete whitelists that consist of the IP addresses allowed to access the databases.

Context

If your business involves multiple applications and you need to add a whitelist for each of them, you can sort the IP addresses into different whitelists.

Create a whitelist

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click in the **Actions** column corresponding to the instance and select **Manage**.

4. In the left-side navigation pane, choose **Data Security > Whitelist Setting**.
5. Click **Add a Whitelist Group** in the upper-left corner of the page.
6. In the panel that appears, configure **Group Name** and **IP White List** and click **OK**.

> **Note**
> - **Group Name**: The name must be 2 to 32 characters in length, and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter and end with a lowercase letter or digit.
> - **IP White List**
>   - Separate multiple IP addresses with commas (,). You can add a maximum of 1,000 different IP addresses to a whitelist. Supported formats are IP addresses such as 0.0.0.0/0 and 10.23.12.24, or CIDR blocks such as 10.23.12.24/24. /24 indicates the length of the IP address prefix in the CIDR block. An IP address prefix can contain 1 to 32 bits.
>   - If the whitelist is empty or only contains 0.0.0.0/0, all devices are granted access. This is risky for your ApsaraDB for MongoDB instance. We recommend that you add only the IP addresses or CIDR blocks of your own web servers to the whitelist.

Delete a whitelist

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, choose **Data Security > Whitelist Setting**.
5. Find the whitelist, and choose ⋮ > **Delete Whitelist Group** in the **Actions** column.

> **Note** You cannot delete the **default** whitelist.

6. In the Delete Whitelist Group message, click **OK**.

16.1.7.3. Audit logs

This topic describes audit logs provided in the ApsaraDB for MongoDB console. You can query the statement execution logs, operations logs, and error logs of an ApsaraDB for MongoDB instance to locate and analyze faults.

Context

The audit log feature records all operations that a client performs on a connected database. This feature provides references for you to perform fault analysis, behavior analysis, and security auditing because you can obtain the operation execution details from the audit logs. Audit logs are essential in the regulatory operations of Finance Cloud and other core business scenarios.

> **Note** Audit logs are stored for seven days, after which they are deleted.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see

Log on to the ApsaraDB for MongoDB console.

2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, choose **Data Security > Audit Logs**.
5. Click **Enable Audit Log** in the upper-left corner. In the Enable Audit message, click **OK**.

Result

On the **Audit Log** page, specify the time range, database name, database user, and keyword to query audit logs. You can also perform the following operations:

- **Export File**: exports an audit log file.
- **File List**: displays a list of audit logs.
- **Disable Audit Log**: stops the collection of information on database operations and deletes the saved audit logs.

16.1.7.4. Configure SSL encryption for an ApsaraDB for MongoDB instance

This topic describes how to enhance link security by enabling Secure Sockets Layer (SSL) encryption and installing SSL CA certificates on your application services. The SSL encryption feature encrypts network connections at the transport layer to improve data security and ensure data integrity during communication.

Prerequisites

- The instance is a replica set instance.
- The MongoDB version of the instance is 3.4, 4.0, or 4.2.

Notes

When you enable or disable SSL encryption or update SSL CA certificates for an instance, the instance is restarted. Plan your operations in advance and make sure that your application is configured to reconnect to the instance after it is disconnected.

> **Note** When an instance is restarted, all its nodes are restarted in turn and each node has a transient connection error of about 30 seconds. If the instance contains more than 10,000 collections, the transient connection error last longer.

Precautions

- You can download SSL CA certificate files only from the ApsaraDB for MongoDB console.
- After you enable SSL encryption for an instance, the CPU utilization of the instance is significantly increased. We recommend that you enable SSL encryption only when necessary. For example, you can enable SSL encryption when you connect to an ApsaraDB for MongoDB instance over the Internet.

  > **Note** Internal network connections are more secure than Internet connections and do not need SSL encryption.

- After you enable SSL encryption for an instance, both SSL and non-SSL connections are supported.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, choose **Data Security > SSL**.
5. Perform one of the following operations.

> **Note** When you enable or disable SSL encryption or update SSL CA certificates for an instance, the instance is restarted. Plan your operations in advance and make sure that your applications can automatically re-establish a connection.

Operation	Prerequisite	Procedure
Enable SSL encryption	The SSL encryption status is **Disabled**.	Turn on **SSL Status**. In the message that appears, click **OK**.
Update an SSL CA certificate	The SSL encryption status is **Enabled**.	Click **Update Certificate**. In the message that appears, click **OK**.
Download an SSL CA certificate file	The SSL encryption status is **Enabled**.	Click **Download Certificate** to download an SSL CA certificate file to your computer.
Disable SSL encryption	The SSL encryption status is **Enabled**.	Turn off **SSL Status**. In the message that appears, click **OK**.

16.1.7.5. Configure TDE for an ApsaraDB for MongoDB instance

This topic describes how to configure Transparent Data Encryption (TDE) for an ApsaraDB for MongoDB instance. Before data files are written to disks, TDE encrypts the data files. When data files are loaded from disks to the memory, TDE decrypts the data files. TDE does not increase the sizes of data files. When you use TDE, you do not need to modify your application that uses the ApsaraDB for MongoDB instance. To enhance data security, you can enable the TDE feature for an instance in the ApsaraDB for MongoDB console.

Prerequisites

The MongoDB version of the instance is 4.0 or 4.2.

> **Note** Before you enable TDE, you can create a MongoDB 4.0 or 4.2 instance to test the compatibility between your application and the database version. You can release the instance after the test is complete.

Notes

- When you enable TDE, your instance is restarted, and your application is disconnected from the instance. We recommend that you enable TDE during off-peak hours and make sure that your application can reconnect to the instance after it is disconnected.
- TDE increases the CPU utilization of your instance.

Precautions

- You cannot disable TDE after it is enabled.

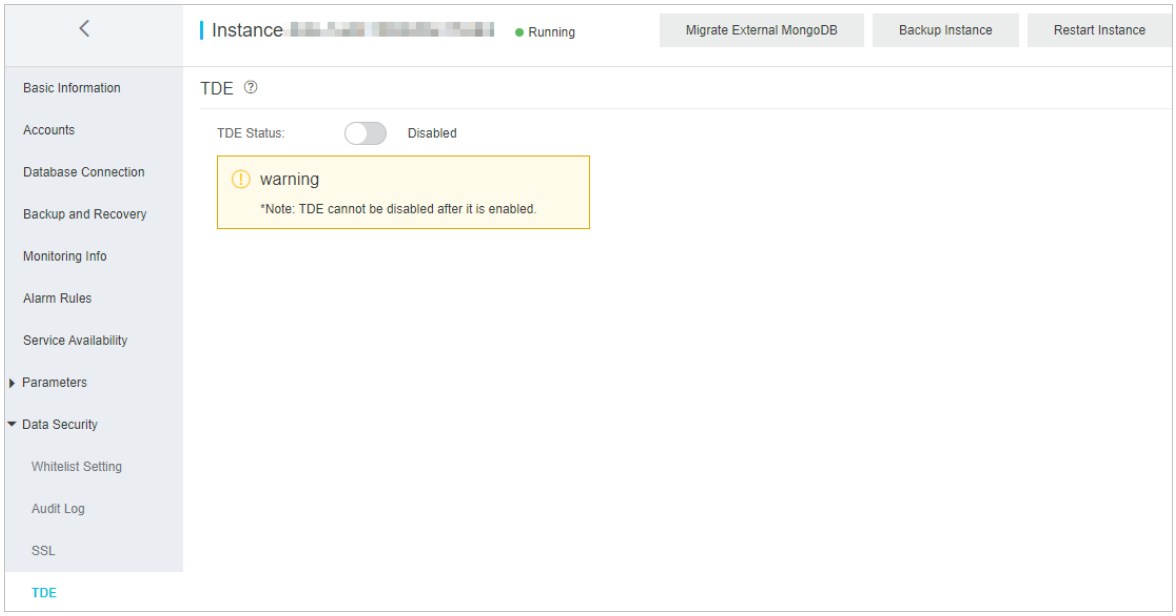
- You can enable TDE for an instance and disable encryption for a collection.

  > **Note** In special business scenarios, you can choose not to encrypt a collection when you create it. For more information, see Disable encryption for a specified collection.

- After you enable TDE, only new collections are encrypted. Existing collections are not encrypted.
- Key Management Service (KMS) generates and manages the keys used by TDE. ApsaraDB for MongoDB does not provide keys or certificates required for encryption.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, choose **Data Security > TDE**.
5. Turn on **TDE Status** to enable TDE.



6. In the **Restart Instance** dialog box, click **OK**.
   The instance status changes to **Modifying TDE**. After the status changes to **Running**, TDE is enabled.

Disable encryption for a specified collection

After you enable TDE, all new collections are encrypted. When you create a collection, you can perform the following steps to disable encryption for the collection:

1. Connect to a replica set instance by using the mongo shell. For more information, see Connect to a replica set instance by using the mongo shell.
2. Run the following command to create a collection and disable the encryption feature:

```
db.createCollection("<collection_name>",{ storageEngine: { wiredTiger: { configString: "encryption=(name=none)" } } })
```

> **Note** <collection_name>: the name of the collection.

Example:

```
db.createCollection("customer",{ storageEngine: { wiredTiger: { configString: "encryption=(name=none)" } } })
```

16.1.7.6. Use the mongo shell to connect to an ApsaraDB for MongoDB database in SSL encryption mode

This topic describes how to use the mongo shell to connect to an ApsaraDB for MongoDB database in Secure Sockets Layer (SSL) encryption mode. SSL encryption can encrypt network connections at the transport layer to improve data security and ensure data integrity.

Prerequisites

- The instance is a replica set instance, and the database version of the instance is 3.4, 4.0, or 4.2.
- The mongo shell of the required version is installed on the local server or ECS instance from which you want to connect to the database. For more information about the installation procedure, visit MongoDB official documentation.
- SSL encryption is enabled for the instance. For more information, see Configure SSL encryption.
- The IP address of the local server or the ECS instance is added to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

Precautions

After you enable SSL encryption for an instance, the CPU utilization of the instance is significantly increased. We recommend that you enable SSL encryption only when necessary.

Procedure

A local server with a Linux operating system is used in the following example.

1. Download an SSL CA certificate package. For more information, see Configure SSL encryption.
2. Decompress the package and upload the certificate files to the local server or the ECS instance where the mongo shell is installed.

   > **Note** In this example, the *.pem* file is uploaded to the */root/sslcafile/* directory of the local server.

3. On the local server or the ECS instance, run the following command to connect to a database of the ApsaraDB for MongoDB instance:

   ```
   mongo --host <host> -u <username> -p --authenticationDatabase <database> --ssl --sslCAFile <sslCAFile_path> --sslAllowInvalidHostnames
   ```

> **Note**
>
> - <host>: the connection string of the primary or secondary node for a replica set instance or of mongos node for a sharded cluster instance. For more information, see Overview of replica set instance connections or Overview of sharded cluster instance connections. If you want to connect to a database of the ApsaraDB for MongoDB instance over an internal network, make sure that the ApsaraDB for MongoDB instance has the same network type as the ECS instance. If the network type is VPC, make sure that the two instances are in the same VPC.
> - <username>: the username you use to log on to a database of the ApsaraDB for MongoDB instance. The default username is root.
> - <database>: the name of database corresponding to the username if authentication is enabled. If the database username is root, enter admin.
> - <sslCAFile_path>: the path of the SSL CA certificate files.

Example:

```
mongo --host dds-bpxxxxxxxx-pub.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticationDatabase
admin --ssl --sslCAFile /root/sslcafile/ApsaraDB-CA-Chain.pem  --sslAllowInvalidHostnames
```

4. When `Enter password:` is displayed, enter the password of the database user and press Enter.

> **Note**
>
> - The password characters are not displayed when you enter the password.
> - If you forget the password of the root user, you can reset it. For more information, see Reset the password for an ApsaraDB for MongoDB instance.

16.1.8. CloudDBA

16.1.8.1. Authorize DAS to manage ApsaraDB for MongoDB instances

Database Autonomy Service (DAS) supports fast scaling, switchover, and centralized management of multiple environments. DAS is integrated into ApsaraDB for MongoDB to facilitate operations and maintenance (O&M). This topic describes how to authorize DAS to manage your ApsaraDB for MongoDB instances when you use the real-time performance, session, and capacity analysis features of CloudDBA for the first time.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, choose **CloudDBA > Realtime performance**.
5. Specify **Database Account** and **Password**, and click **Authorize**.

Result

The page is refreshed. Then, you can use the real-time performance, session, and capacity analysis features. You

will not be prompted for authorization again.

16.1.8.2. Performance trends

This topic describes how to view performance trends in specific ranges, compare performance trends, and customize charts to view performance trends on your ApsaraDB for MongoDB instances.

Go to the Performance page

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, choose **CloudDBA > Performance Trends**.

> **Note** For more information about performance trends, see relevant topics in Database Autonomy Service (DAS) User Guide.

16.1.8.3. Real-time performance

This topic describes how to view real-time monitoring statistics of your ApsaraDB for MongoDB instances, such as read/write latency, queries per second (QPS), operations, connections, and network traffic.

Prerequisites

Database Autonomy Service (DAS) is authorized to manage ApsaraDB for MongoDB instances. For more information, see Authorize DAS to manage ApsaraDB for MongoDB instances.

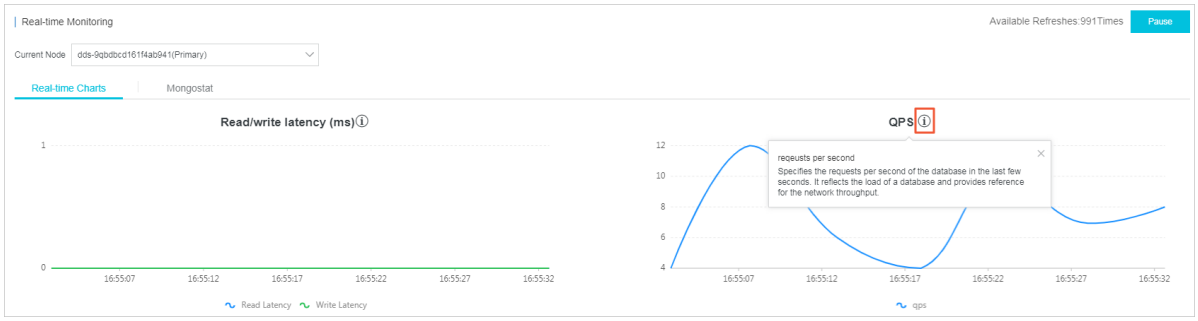
Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, choose **CloudDBA > Realtime performance**.

Real-time Monitoring

On the Real-time Monitoring page, you can select the Real-time Charts or Mongostat tab to view monitoring statistics. When you refresh or go to the **Real-time Monitoring** page, the information on the Real-time Charts and Mongostat tabs is refreshed, and the number of available refreshes is reset in the upper-right corner.

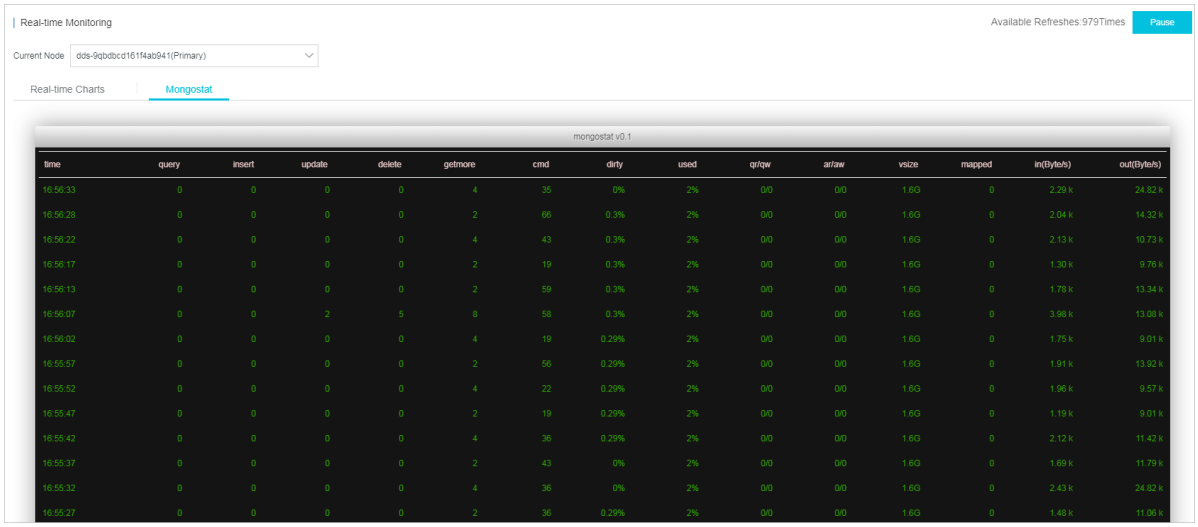
- Real-time Charts



The **Real-time Charts** tab is displayed on the Real-time Monitoring page. Line charts on the tab are refreshed every five seconds.

> **Note** For more information about individual metrics, click the icon above each chart.

- mongostat



Click the **Mongostat** tab. On the tab, you can view Mongostat command outputs. A new line of monitoring data is added every five seconds. The tab can contain up to 999 lines of information.

> **Note** For more information about Mongostat command outputs, visit MongoDB official documentation.

16.1.8.4. Instance sessions

This topic describes how to view real-time monitoring statistics of your ApsaraDB for MongoDB instances, such as read/write latency, queries per second (QPS), operations, connections, and network traffic.

Prerequisites

Database Autonomy Service (DAS) is authorized to manage your ApsaraDB for MongoDB instances. For more information, see Authorize DAS to manage ApsaraDB for MongoDB instances.

View instance sessions

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.

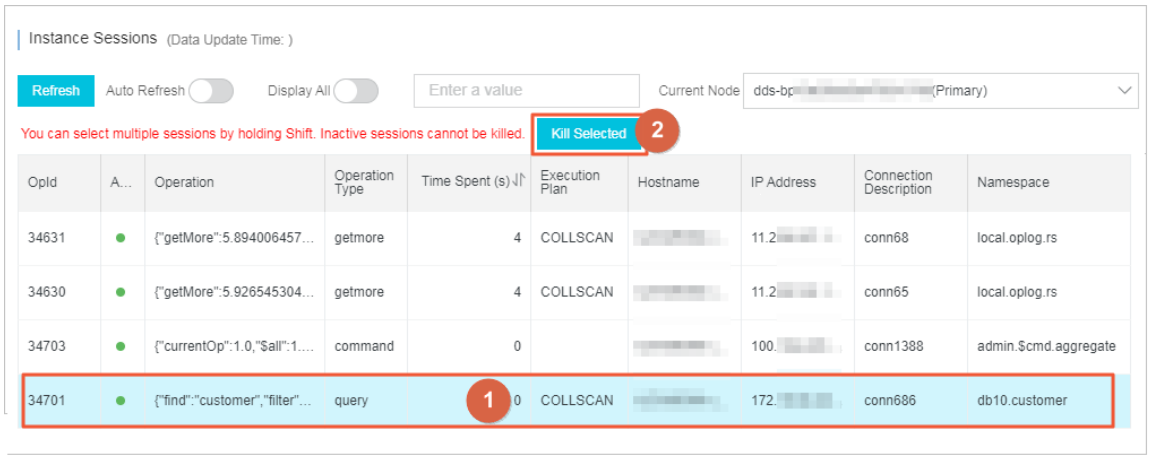
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, choose **CloudDBA > Session**.

> **Note**
> - If you turn on **Auto Refresh**, the system updates session data on the page every 30 seconds.
> - By default, the system displays only active sessions. You can turn on **Display All** to view both active and inactive sessions.
> - In the **Session Statistics** section, you can view information about sessions in the **Overview**, **Statistics by Client**, and **Statistics by Namespace** charts.

Terminate instance sessions

> **Warning** To avoid unexpected results, we recommend that you do not terminate system-level sessions.

1. In the **Instance Sessions** section, select the sessions you want to terminate and click **Kill Selected**.



2. In the message that appears, click **OK**.

16.1.8.5. Capacity analysis

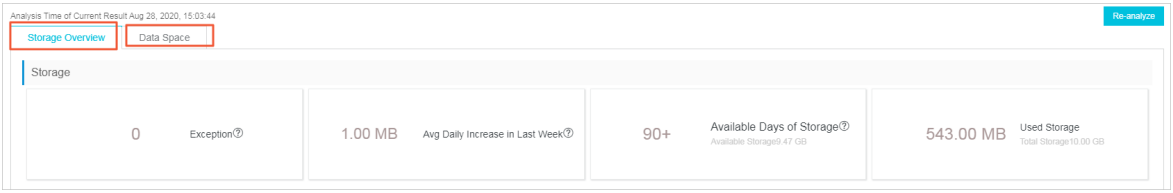
This topic describes how to view information about the capacity analysis feature, including **storage**, **exceptions**, **storage trend**, **tablespaces**, and **data space**. The information helps you detect and resolve exceptions in the database storage to ensure database stability.

Prerequisites

Database Autonomy Service (DAS) is authorized to manage ApsaraDB for MongoDB instances. For more information, see Authorize DAS to manage ApsaraDB for MongoDB instances.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, choose **CloudDBA > Capacity analysis**.
5. In the upper-right corner, click **Re-analyze**. Then, wait until the analysis is complete.
6. On the **Storage Overview** or **Data Space** tab, view the analysis results.



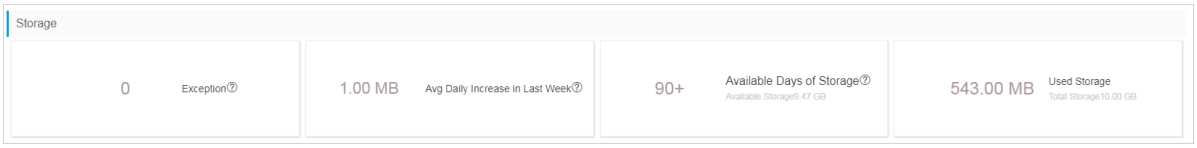
For more information about Storage Overview, see Storage Overview.

For more information about Data Space, see Data Space.

Storage Overview

On the Storage Overview tab, you can view information in the **Storage**, **Exceptions**, **Storage Trend**, and **Tablespaces** sections.

- **Storage Overview**

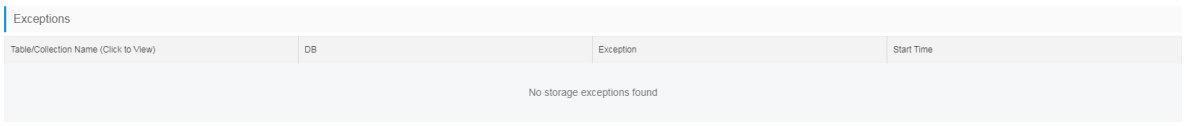


Item	Description
Exceptions	The number of detected storage exceptions. ApsaraDB for MongoDB can detect the following types of exceptions: ○ Over 90% of the storage capacity is used. ○ The total physical storage will be unavailable in seven days. ○ The number of indexes in a collection exceeds 10.
Avg Daily Increase in Last Week	The average daily increase of storage usage over the last seven days. Formula: (Storage usage at the time of collection - Storage usage seven days ago)/7. **Note** ○ The increase speed is the average value during the seven days before the collection time. ○ This parameter is only used as a reference for scenarios in which the traffic remains stable. Abrupt storage changes caused by batch imports, deletion of historical data, instance migration, or instance re-creation affect the accuracy of the data.

Item	Description
Available Days of Storage	The estimated number of days during which storage space is available. Formula: Size of available storage space/Average daily increase over the last week. **Note** ○ 90+ indicates that the disk storage is sufficient for more than 90 days of usage. ○ This parameter is used only as a reference for scenarios in which the traffic remains stable. Abrupt storage changes caused by batch imports, deletion of historical data, instance migration, or instance re-creation affect the accuracy of the data.
Used Storage	The size of used storage space in contrast to the total size of storage space.

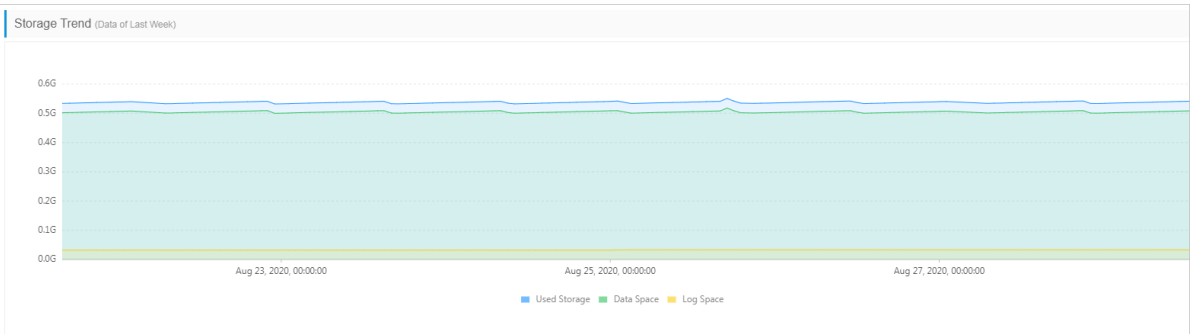
- **Exceptions**

  Information about detected storage exceptions. You can resolve the exceptions based on information in this section.



- **Storage Trend**

  Changes of storage usage over the last week, such as changes of **used storage**, **data space**, and **log space**.



- **Tablespaces**

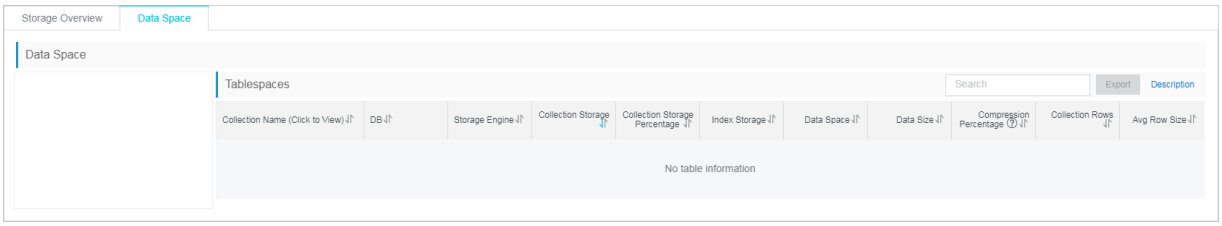
  Information of all tables, such as the database name, storage engine, and collection storage.

Note You can click a collection name to view its indexes.

Data Space

The Data Space tab shows the total storage capacity and tablespace information of each database.

> **Note**
> - You can click a data space to view its tablespace information.
> - You can click a collection name to view its indexes.



16.1.8.6. Slow query logs

This topic describes how to view slow query logs of ApsaraDB for MongoDB instances. You can locate, analyze, diagnose, and track slow query logs to create indexes, which improves the utilization of resources in the instances.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, choose **CloudDBA > Slow query log**.

   > **Note** By default, slow query logs generated in the past 15 minutes are displayed in the trend chart. You can specify the time range and click **Search** to view slow query logs in specific periods of time. The maximum time range is one day.

5. View details of slow query logs by using one of the following methods:

   Method 1:

   i. Click the **Slow Log Details** tab in the lower part of the page.

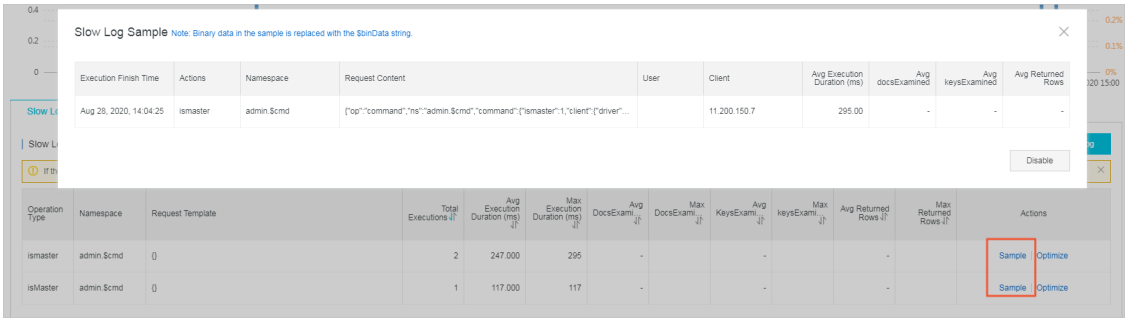
   ii. On the **Slow Log Details** tab, select the database that you want to query.

   > **Note** If the request content of the target database is hidden, you can move the pointer over the corresponding **request content** and view the complete content.

   Method 2:

   i. In the slow log trend chart, click the time of a specific slow query log and view its details on the **Slow Log Statistical** tab.

ii. On the **Slow Log Statistical** tab, click **Sample** in the **Actions** column.
In the **Slow Log Sample** dialog box, you can view details of the slow query log.



Note If the request content of the target database is hidden, you can move the pointer over the corresponding **request content** and view the complete content.

Export slow query logs

You can click **Export Slow Log** on the **Slow Log Statistical** tab to save the slow query log information to your computer.

17.ApsaraDB for OceanBase

17.1. 文档标题缺失，请补全后重新导出

17.1.1. What is ApsaraDB for OceanBase?

ApsaraDB for OceanBase is a financial-grade, distributed relational database service that features high performance, high availability, and high scalability. It supports active geo-redundancy and geo-disaster recovery to ensure high availability. It also supports high scalability to meet the increasing business requirements.

These features of ApsaraDB for OceanBase help you handle the challenges that are brought by rapid business growth. ApsaraDB for OceanBase also provides scalable and low latency database services in high throughput scenarios. This ensures improved user experience. For example, during the Double 11 in 2017, ApsaraDB for OceanBase handled all the transactions and payment requests. The maximum number of transactions that were made on Alipay reached 256,000 per second. The maximum number of processed requests per second reached 42 million. ApsaraDB for OceanBase accelerates the development of Internet finance.

The distributed engine of ApsaraDB for OceanBase uses the Paxos protocol and maintains multiple replicas. For the Paxos protocol, transactions can be committed only after they are approved by a majority of the acceptors. The Paxos protocol and multiple-replica design allow ApsaraDB for OceanBase to offer high availability and disaster recovery capabilities. ApsaraDB for OceanBase can help you achieve zero downtime. ApsaraDB for OceanBase supports high-availability architectures, such as active geo-redundancy and geo-disaster recovery. You can deploy the ApsaraDB for OceanBase service across data centers, regions, or continents. ApsaraDB for OceanBase provides financial-grade availability features and ensures strong consistency of transactions.

ApsaraDB for OceanBase is similar to an in-memory database and adopts a read/write splitting architecture. To ensure high efficiency for the storage engine, ApsaraDB for OceanBase stores baseline data in solid-state drives (SSDs) and stores incremental data in memory. This ensures that ApsaraDB for OceanBase offers high performance services. ApsaraDB for OceanBase is a cloud-based database service that supports multi-tenant data isolation. Each cluster of ApsaraDB for OceanBase can provide services for multiple tenants. The tenants are isolated so that they are not affected by each other.

ApsaraDB for OceanBase is compatible with most of the MySQL 5.6 features. This allows you to migrate MySQL-based services to ApsaraDB for OceanBase based on zero or small code modifications. This improves the efficiency of developing applications and migrating services. In ApsaraDB for OceanBase, you can create partitioned tables and use subpartitions. This serves as an alternative to MySQL sharding solutions. The ApsaraDB for OceanBase console provides an easy way for you to manage complex databases. For example, you can use the console to upgrade or downgrade instances, view performance data, and view optimization suggestions.

17.1.2. Quick start

17.1.2.1. Log on to the ApsaraDB for OceanBase O&M console

This topic describes how to use the Apsara Uni-manager Operations Console to log on to the ApsaraDB for OceanBase O&M console. The Google Chrome browser is used in an example in this topic.

Prerequisites

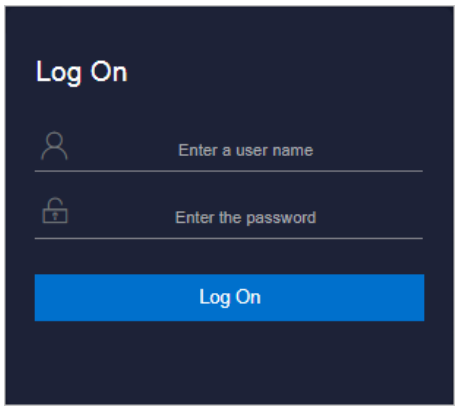
- The endpoint of the Apsara Uni-manager Operations Console and the username and password used to log on to the console are obtained from the deployment personnel or an administrator.

  The endpoint of the Apsara Uni-manager Operations Console is in the following format: *region-id*.ops.console.*intranet-domain-id*.

- A browser is available. We recommend that you use Google Chrome.

Procedure

1. In the address bar, enter the URL *region-id*.aso.*intranet-domain-id*.com and press the Enter key.



> **Note** You can select a language from the drop-down list in the upper-right corner of the page.

2. Enter your username and password.

> **Note** Obtain the username and password for logging on to the ASO console from the deployment personnel or an administrator.

When you log on to the ASO console for the first time, you must change the password of your username as prompted.

To enhance security, a password must meet the following requirements:

- It must contain uppercase and lowercase letters.
- It must contain digits.
- It must contain special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs ($), and percent signs (%).
- It must be 10 to 20 characters in length.

3. Click **Log On** to go to the **ASO** console.
4. In the left-side navigation pane, click **Product Management**. Then, choose **Products > Database Services** and click **OceanBase Cloud Platform**. You are directed to the ApsaraDB for OceanBase O&M console.

17.1.2.2. Log on to the ApsaraDB for OceanBase console

This topic describes how to use the Apsara Uni-manager Management Console to log on to the ApsaraDB for OceanBase console. The Google Chrome browser is used in an example in this topic.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- A browser is available. We recommend that you use the Google Chrome browser.

Context

If you use the Apsara Uni-manager Management Console to log on to the ApsaraDB for OceanBase console, you are granted with only the permissions to view monitoring data and manage instances. If you need to perform operations and maintenance (O&M) tasks on clusters, follow the instructions described in Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase and perform O&M tasks based on your business needs.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

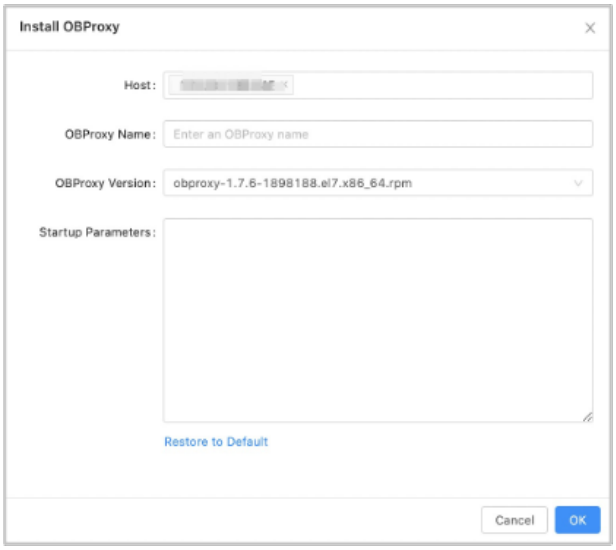
3. Click **Log On**.
4. In the top navigation bar, click **Products** and choose **Database Services > ApsaraDB for OceanBase**.
5. Select an organization from the **Organization** drop-down list and select a region from the **Region** drop-down list. Then, click **OceanBase Cloud Platform**. You are directed to the ApsaraDB for OceanBase console.

17.1.2.3. Install OBProxy

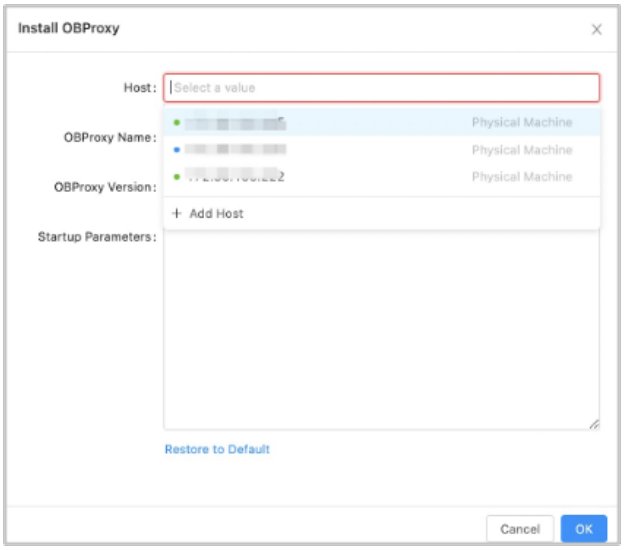
In the upper-right corner of the page, click Return to Old Version to go to the OBProxy page and install OBProxy.

Procedure

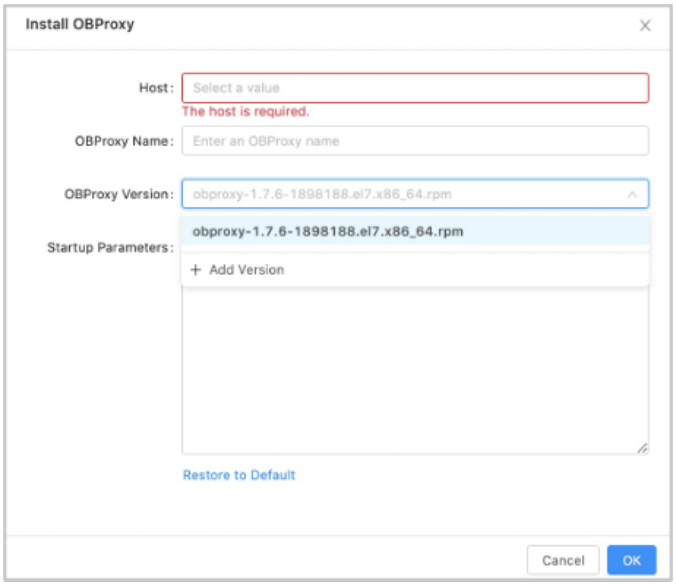
1. In the navigation pane, choose **Maintenance** to go to the **OBProxy** page.
2. Click **Install OBProxy**.
3. In the dialog box that appears, specify the following parameters:



- **Host**: Select a host from the drop-down list, or click **Add Host** to add a new host.



- OBProxy Version: Select an OBProxy version from the drop-down list, or click **Add Version** and click **Upload** in the dialog box that appears, to upload the corresponding OBProxy file.



4. Click **OK**. This will generate an OBProxy operations task. You can choose **System Management > Tasks** to check the installation progress.

17.1.2.4. Create a cluster

This topic describes how to create a cluster as needed.

Prerequisites

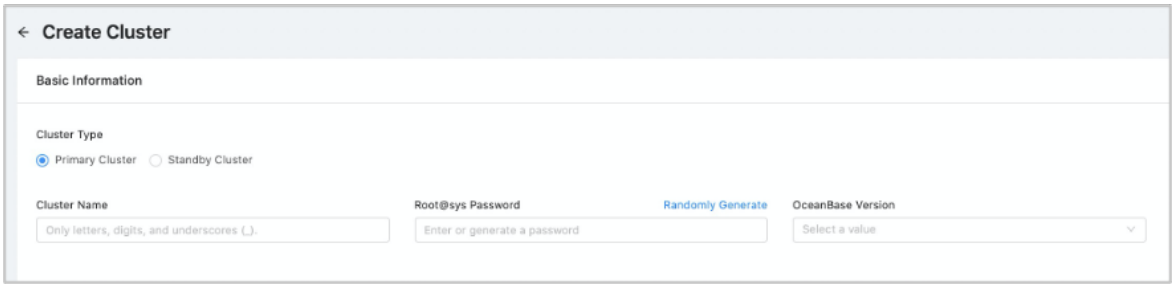
The permissions to create a cluster are granted to the current user.

Procedure

1. After you log on to OCP, find the entry to create a cluster based on your actual business scenario.
   - If you do not have a cluster to manage, the system prompts you to create a cluster on the **Cluster Overview** page. In the dialog box that appears, click **Create Cluster**.

- If you have a cluster to manage, click **Create Cluster** in the upper-right corner of the **Cluster Overview** page.

2. On the **Create Cluster** page, configure the basic cluster information.



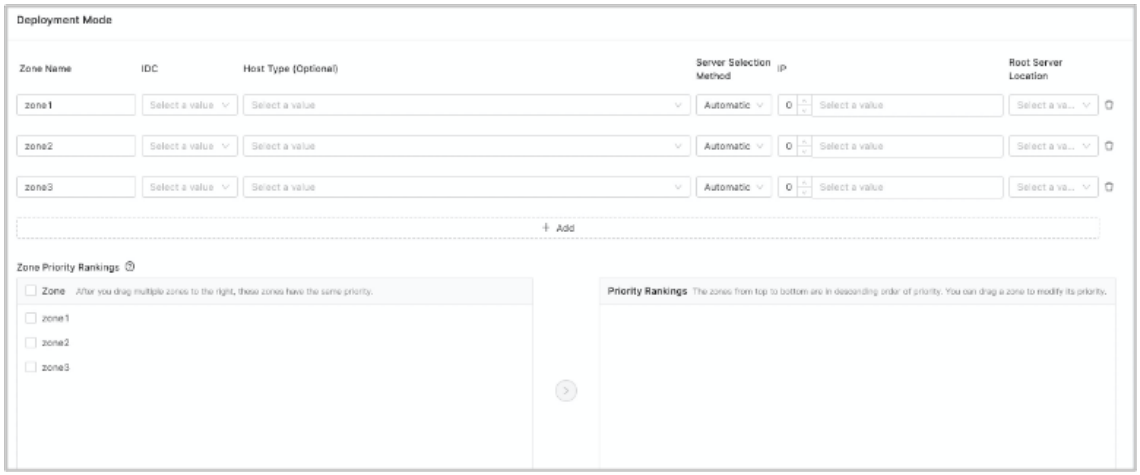
The following table describes the parameters related to the basic cluster information.

Parameter	Description
Cluster Type	You can select **Primary Cluster** or **Standby Cluster**. Before you select **Standby Cluster**, make sure that a primary cluster is created.
Cluster Name	The name of the cluster that you want to manage. The cluster name must be 2 to 48 characters in length, and can contain letters, digits, and underscores (_). It must start with a letter.
Root@sys Password	You can set a custom password or use a password that is generated by the system in a random way. The password must meet the following complexity requirements: • The password must be 8 to 32 characters in length. • The password must contain at least two digits, two uppercase letters, two lowercase letters, and two special characters. The following special characters are supported: . _+@#$%)
OceanBase Version	You can select an existing version from the drop-down list, or click **Add Version** at the bottom of the drop-down list to upload a version.

3. Configure the deployment mode for the cluster.

By default, you can add three zones. To add four or more zones in the cluster, click **Add** at the lower part of the field.

To deploy the cluster to one or two zones, click the Delete icon next to the zone that you want to delete.



The following table describes the parameters for a zone.

Parameter	Description
Zone Name	In most cases, a default name is generated. You can customize the name as needed. The zone name must be 2 to xx characters in length, and can contain letters, digits, and underscores (_). It must start with a letter.
IDC	The data center to which the zone belongs. Ensure that the zones you specify for the cluster belong to the same data center.
Host Type	This parameter is optional. If you select a host type, the hosts are filtered based on the host type.
Server Selection Method	You can select **Automatic** or **Manual**.
IP	You can select multiple IP addresses. If you select **Automatic** from the **Server Selection Method** drop-down list, you only need to specify the number of IP addresses. OCP selects the specific number of available servers in an automatic way. If you select **Manual** from the **Server Selection Method** drop-down list, you must manually select multiple IP addresses from the drop-down list.
Root Server Location	You can select an IP address for the root server.

Parameter	Description
Priority Rankings	The priority rankings of the zones. This parameter specifies the priority of each zone to be selected as the primary zone on the system tenant. The list on the left displays all the zones in the current cluster. You can select one or more zones from the left-side list and add them to the right-side list. By default, a zone that you select first has a higher priority than a zone that you select later. If you select multiple zones at a time, they have the same priority. After you move zones to the right-side list, you can drag a zone to change its priority in the right-side list. The zones from top to bottom are in descending order of priority.

4. After you configure the preceding parameters, click **Submit**.
5. In the **Confirm Information** dialog box, confirm the information and click **OK**.

17.1.2.5. Create a tenant

A tenant is a container for various database objects and resources such as CPU, memory, and I/O. You can create tenants in a cluster based on your business requirements.

You can use one of the following two methods to create a tenant:

- Create a tenant on the Tenant Overview page.
- Create a tenant on the Tenant Management page of a specified cluster.

This topic describes how to create a tenant on the Tenant Management page of a specified cluster.

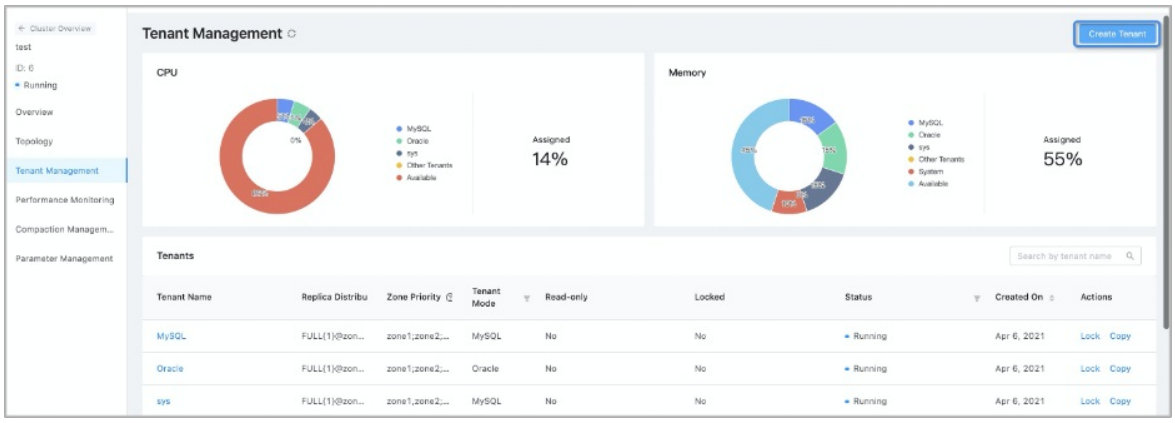
Prerequisites

- The cluster where you create a tenant must be a primary cluster and must be in running state.
- The current user you log on to is the system administrator, OCP tenant administrator role, or other roles that have permissions to manage the cluster.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster where you want to create a tenant.
3. In the left-side navigation pane of the page that appears, click **Tenant Management**.

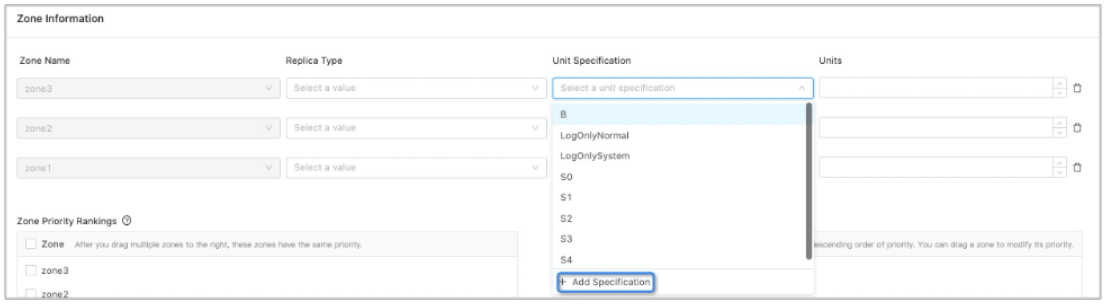
4. In the upper-right corner, click **Create Tenant**.



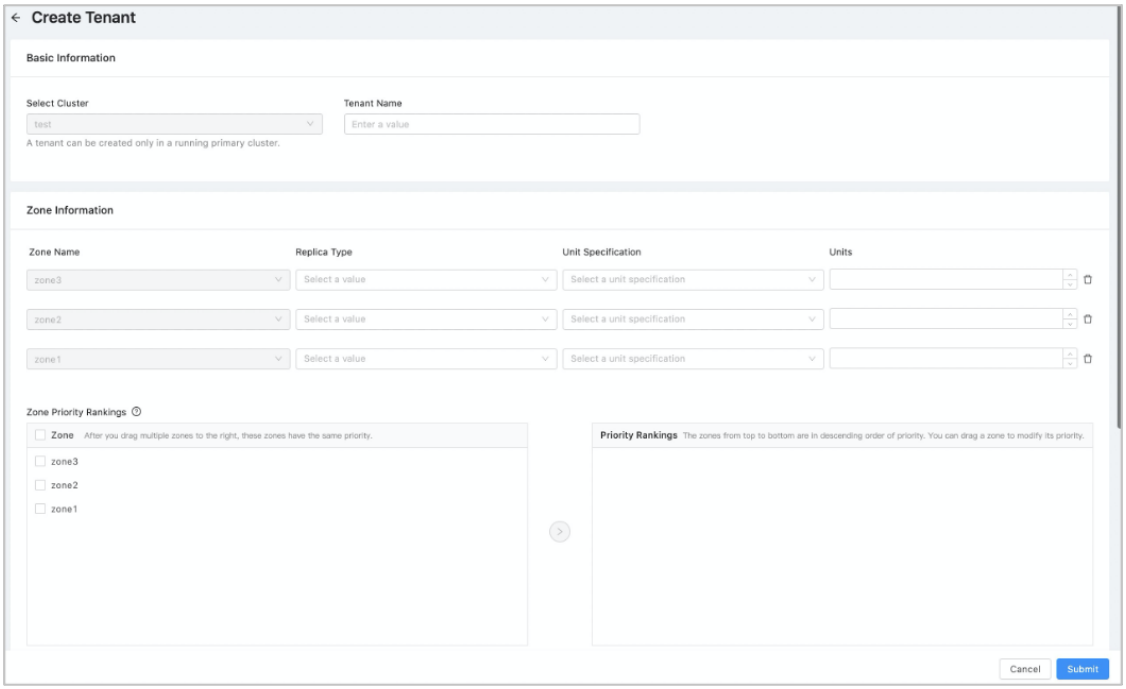
5. Configure **Basic Information**.
   i. The default cluster is the current cluster.
   ii. Enter a **Tenant Name**. The tenant name must be 2 to 64 characters in length and can contain letters, digits, and underscores (_).
6. Configure **Zone Information**.
   i. Specify the Replica Type, Unit Specification, and Units for zone 1, zone 2, and zone 3.

      After you select a cluster, the page shows a list of zones that can be configured based on the zone information of the selected cluster. For zones that do not store data replicas, you can delete the zone by clicking the delete icon on the rightmost side.

      - You can select Full-featured Replica, Read-only Replica, and Log Replica from the Replica Type drop-down list.
      - OCP provides a set of built-in Unit specifications. You can also click Add Specification at the bottom of the Unit Specification drop-down list to add custom specifications.
      - Specify the number of units in the zone. Note that the number of units cannot exceed the number of servers in the zone.



ii. Configure the **Zone Priority Rankings**. If you drag multiple zones to the right, these zones have the same priority.



7. Configure the **Basic Settings**.

i. Settings

- If the tenant mode is MySQL, its administrator account is "root".
- If the tenant mode is Oracle, its administrator account is "SYS". The password must be 8 to 32 characters in length and contain at least two digits, two uppercase letters, two lowercase letters, and two special characters. Special characters include: _+@#$%.

ii. Select a **Tenant Mode**.

- You can select Oracle or MySQL.
- The Oracle tenant mode is supported only when the OceanBase version of the selected cluster is 2.1 or later.

iii. Configure the **Character Set and Encoding**.

- If you select the MySQL tenant mode, the following character sets are available: utf8mb4, binary, gbk, and gb18030. Default value: utf8mb4.
- If you select the Oracle tenant mode, the following character sets are available: utf8mb4, gbk, and gb18030. Default value: utf8mb4.

iv. Optional. Configure **Remarks**.

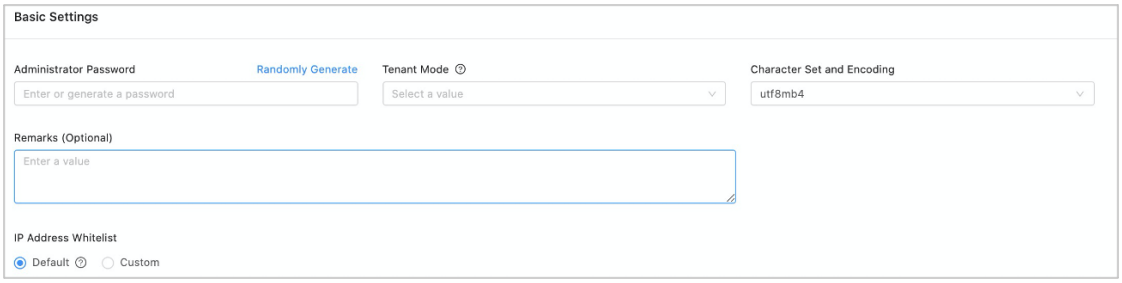
v. Configure the **IP Address Whitelist**.

- You can specify the clients that the tenant can log on to. If you do not specify, the default configuration is "%". It indicates that the tenant can log to all clients. When you customize the whitelist, note that the whitelist must include the IP addresses of the OCP server and the OBProxy on which the OCP server depends. Otherwise, OCP cannot manage this tenant.
- Default: Accesses from all IP addresses is supported.
- Custom: Configure an IP address whitelist. Only accesses from the IP addresses in the whitelist are supported.

- Description of whitelist format:

- IP addresses. For example, 10.10.10.10,10.10.10.11.
- Subnet mask. For example, 10.10.10.0/24.
- Fuzzy match. For example, 10.10.10. % or 10.10.10. _
- A mix of multiple formats. For example, 10.10.10.10,10.10.10.11,10.10.10. %,10.10.10. _,10.10.10.0/24

> **Note**
> The percent sign (%) indicates that all clients can connect to the tenant.



8. Click **Submit**.

17.1.3. OceanBase Cloud Platform

17.1.3.1. Clusters management

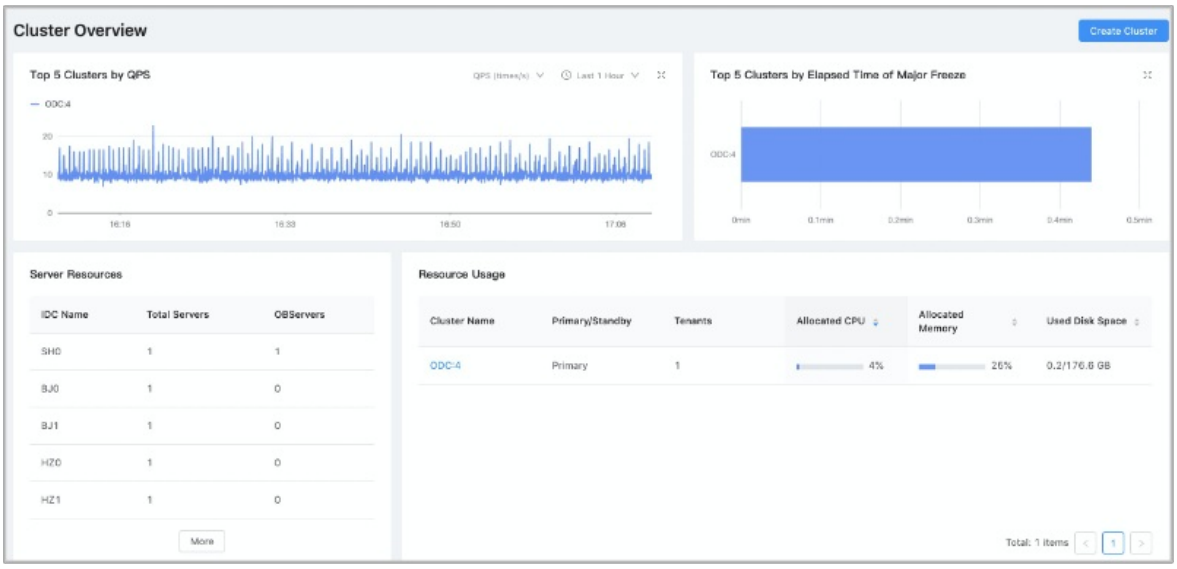
17.1.3.1.1. Overview

OCP provides full lifecycle management for ApsaraDB for OceanBase clusters. For example, you can create, monitor, configure, upgrade, or delete clusters on OCP. You can also perform O&M operations on clusters.

17.1.3.1.2. Cluster overview

This topic describes the information you can view on the Cluster Overview page.

After you log on to OCP, the **Cluster Overview** page appears. To view the overview data of a cluster, the current logon user must have the permissions to manage the cluster or the read-only permissions on the cluster.



The **Cluster Overview** page displays the information of the cluster for the current logon user. The information includes **Top 5 Clusters by QPS**, **Top 5 Clusters by Elapsed Time of Major Freeze**, **ServerResources**, **Resource Usage**, and **Clusters**.

Top 5 clusters by QPS

The **Top 5 Clusters by QPS** section shows the top five clusters in terms of the average value of a performance metric within a specific time period.

You can select the performance metric you want to view and the corresponding time range based on your business needs.

You can select the following performance metrics:

- QPS (times/s)
- Query response time (us)
- Number of active sessions
- CPU usage (%)

You can select the following time ranges:

- Last 1 hour
- Last 24 hours
- Last 7 days

Top 5 clusters by elapsed time of major freeze

The **Top 5 Clusters by Elapsed Time of Major Freeze** section shows the top five clusters in terms of the average elapsed time of major freezes in the last three days in a column chart. Major freeze operation allows you to merge dynamic data with static data, but it is time-consuming. If the amount of incremental data generated by minor freezes exceeds a specified threshold, a major freeze operation is triggered.

If no clusters are merged within the last three days, this section is empty.

Server resources

The **Server Resource** section displays the total number of servers and the number of OBServer deployed in each data center. This section can provide a reference for cluster scaling.

Resource usage

The **Resource Usage** section displays the name and type of every cluster, and the number of tenants in a cluster. The section also displays the unit and unit numbers of allocated resources in a cluster, including the percentage of allocated CPU and memory, used disk space, and total disk space.

Clusters

The **Clusters** section displays the name, ID, OceanBase version, deployment mode, state, creation time, and alert information of each cluster. The deployment mode indicates the region where each zone is deployed and the number of OBServers

By default, only the information of primary clusters is displayed. Click the icon on the left side to view the information of standby clusters.

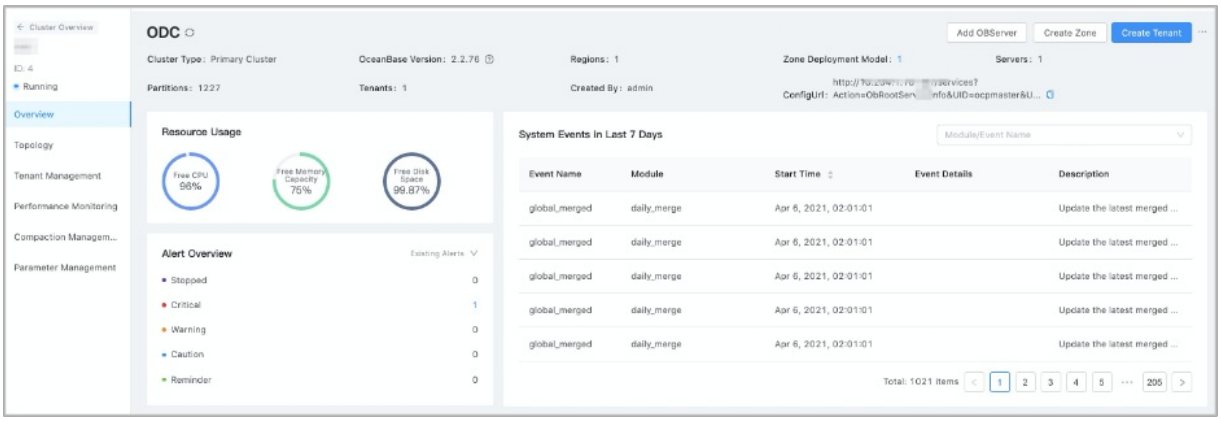
You can click the name of a cluster to go to the cluster details page.

17.1.3.1.3. Manage a single cluster

17.1.3.1.3.1. Cluster overview

You can select a cluster on the cluster overview page to view the overall information of the cluster. You can view the summary information, resource usage, alert overview information, system events of the last seven days, and the zone and OBServer lists.

In the **Clusters** section of the **Cluster Overview** page, find the cluster you want to view and click its cluster name. The **Overview** page for the cluster appears.



The cluster overview information

The Overview page displays the following information of the cluster: cluster type, OceanBase version, the creator, configUrl and the number of regions, zones, servers, partitions, and tenants.

Resource usage

The Resource Usage section displays the percentage of the following available system resources: free CPU, free memory capacity, and free disk space.

Alert overview

The Alert Overview section displays the information of all alerts related to the current cluster. The alert information is classified and displayed by alert severity.

Currently, the alerts are classified into the following five types: stopped, critical, warning, caution, and reminder.

By default, various existing alerts are displayed. You can view the alerts generated in the last 24 hours, the last 7 days, or the last 30 days based on business requirements.

System events in the last 7 days

This section displays the event details related to the RootService of the current cluster, and all events are sorted by the start time of the event.

The displayed information of each system event includes the event name, module, start time, event details, and event description.

The system displays event details and description for common events and does not display event details and description for uncommon events.

Zones

The Zones section displays the following basic information of each zone in the current cluster: Zone Name, Region, IDC, Root Server, and Status. It also provides a zone-level operation and maintenance (O&M) portal that allows you to add OBServers, and restart, stop, and delete the OBServer process of all nodes in a zone.

OBServers

The OBServers section displays the basic information of each OBServer in the current cluster, including the IP address, the port, the data center, the host type. This section also displays the zone where an OBServer resides and the available resources such as CPU, memory, and disk space.

In addition, OCP provides portals for common O&M operations on each OBServer in the current cluster to facilitates basic O&M operations. These portals allow you to restart, start, stop, replace, and delete OBServers.

17.1.3.1.3.2. Manage clusters

Create a standby cluster

If you have a primary cluster, you can create a standby cluster for the primary cluster.

Prerequisites

Make sure the current logon user is granted the permissions for managing clusters

Procedure

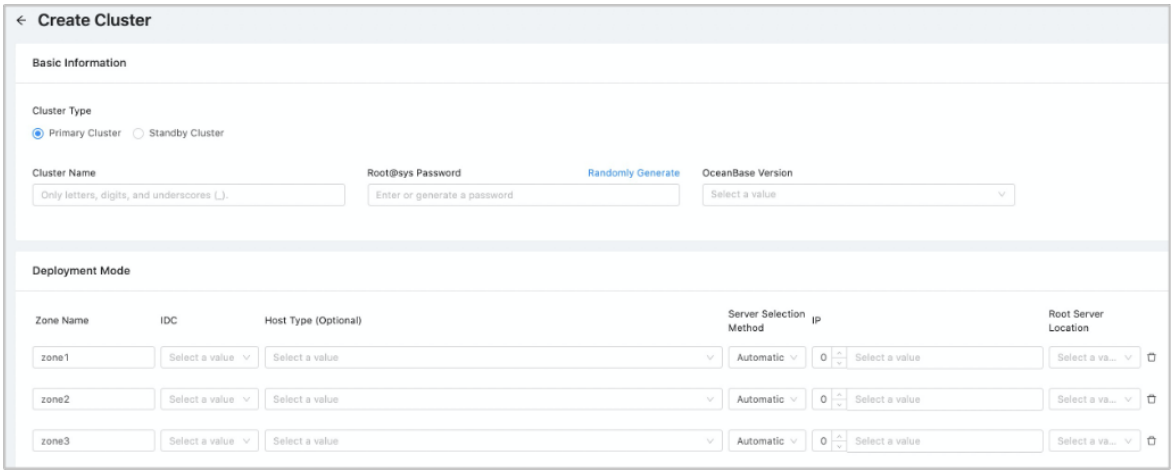
1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. You can use one of the following two methods to go to the page for creating a standby cluster.
   - In the upper-right corner of the **Cluster Overview** page, click **Create Cluster**. On the **Create Cluster** page, set **Cluster Type** as

     Standby Cluster, and select a primary cluster from the **Primary Cluster** drop-down list.
   - In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster to which you want to add a standby cluster.

     In the upper-right corner of the **Overview** page, choose **Create Standby Cluster**.
3. On the **Create Cluster** page, set the deployment mode of the standby cluster.

   By default, you can add three zones. If you want to create more than three zones for the cluster, click **Add** at the lower part of the field.

   If you want to deploy less than three zones, click the delete icon next to the zone that you want to delete.



The following table describes the parameters you must configure for a zone.

Parameter	Description
Zone Name	In most cases, a default name is generated. You can customize the zone name based on your business requirements.

Parameter	Description
IDC	The data center to which the zone belongs. The zones you create must be deployed in the same data center.
Host Type	Optional. The IP addresses are filtered based on the host type you select.
Server Selection Method	You can select **Automatic** or **Manual**.
IP	You can select multiple IP addresses. If you select **Automatic** from the **Server Selection Method** drop-down list, you only need to specify the number of IP addresses. OCP automatically selects the specific number of available physical servers. If you select **Manual** from the **Server Selection Method** drop-down list, you must manually select several IP addresses from the IP drop-down list.
Root Server Location	You can select an IP address as the server where the root server is located.

v. Click **Submit**.

vi. In the **Confirm Information** message, check the information and click **OK**.

Upgrade the version of a cluster

When a new cluster version is available, you can upgrade the version for the cluster that you manage.

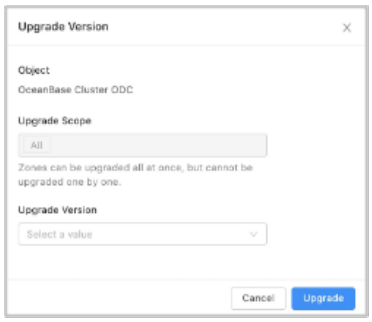
Prerequisites

You have the permissions to manage the cluster.

Procedure

1. Log on to OceanBase Cloud Platform (OCP). By default, the **Cluster Overview** page appears after you log on to OPC.
2. In the **Clusters** section of the **Cluster Overview** page, select the cluster that you want to manage, and then click the cluster name.
3. Click the icon in the upper-right corner of the **Overview** page and select **Upgrade Version**.

4. In the **Upgrade Version** dialog box, select the version to be upgraded.



If the version to be upgraded is not uploaded to OCP, you can click **Add Version** in the lower part of the **Upgrade Version** list to upload the related package.

5. Click **Upgrade**.
6. In the **Confirm Upgrade Path** dialog box, click **OK**.

Change a password

You can change the password of the root user under the cluster system tenant based on your needs.

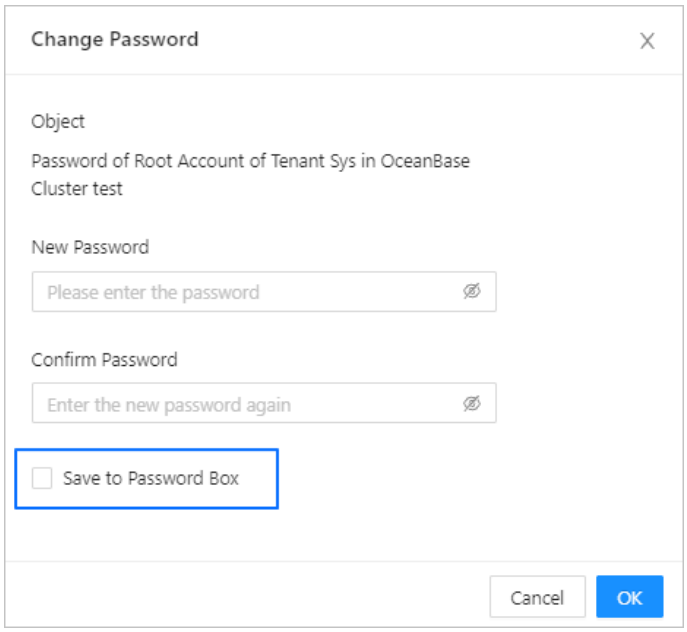
Prerequisites

You have the permissions to manage the cluster.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster** page, click the name of the cluster that you want to manage.
3. In the upper-right corner of the **Overview** page, click the More icon and select **Change Password**.
4. In the **Change Password** dialog box, enter and confirm the new password.

> **Note**
>
> By default, the **Save to Password Box** check box is selected. In this case, when you log on to OCP next time, you can use the new password to log on to the cluster.



5. Click **OK**.

Restart clusters

You can restart the OBServer processes of all nodes in a cluster or restart the OBServer processes in specific zones based on your business needs.

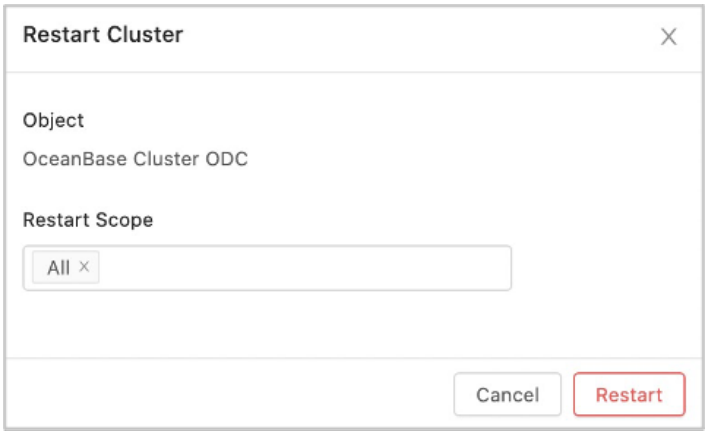
Prerequisites

Make sure the current logon user is granted the permissions for managing clusters

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the upper-right corner of the **Overview** page, click the More icon and select **Restart Cluster**.
4. In the **Restart Scope** field, you can select **All**, a specific zone, or several zones. Then, click **Restart**.

If you select **All**, the system restarts the OBServer processes of all nodes in the cluster. If you select some zones, the system restarts the OBServer processes of these zones.



Stop a cluster

You can stop a cluster based on your business needs. Stopping a cluster stops the OBServer processes running on all nodes in the cluster.

Prerequisites

Make sure the current logon user is granted the permissions for managing clusters.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to stop.
3. In the upper-right corner of the **Overview** page, click the icon and select **Stop Cluster**.

   > **Notice**
   >
   > Stopping a cluster stops the OBServer processes on all nodes in the cluster. Proceed with caution.

4. In the message, click **Stop**.

Delete a cluster

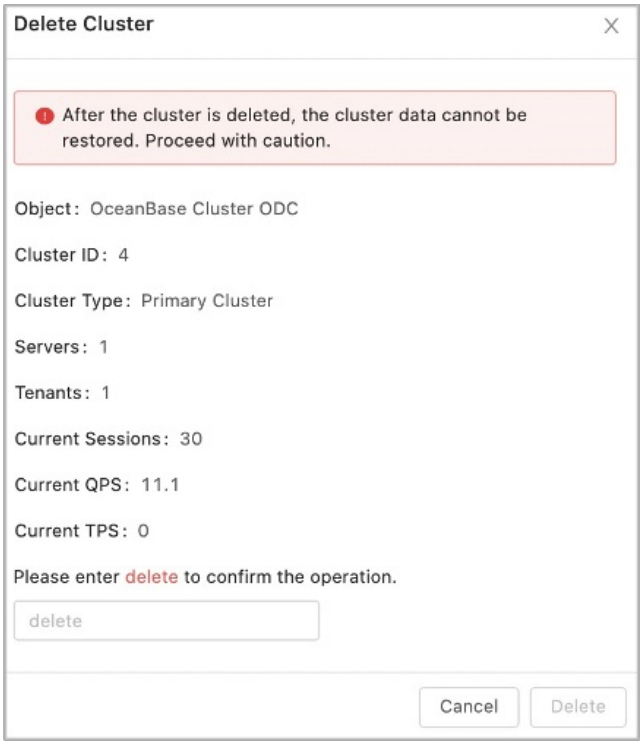
You can delete ApsaraDB for OceanBase clusters based on your needs.

Prerequisites

Permissions to manage the cluster are granted to the current user.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to delete.
3. In the upper-right corner of the **Overview** page, click the iconand select **Delete Cluster**.
4. In the **Delete Cluster** dialog box, enter delete and click **Delete**.



17.1.3.1.3.3. Manage zones

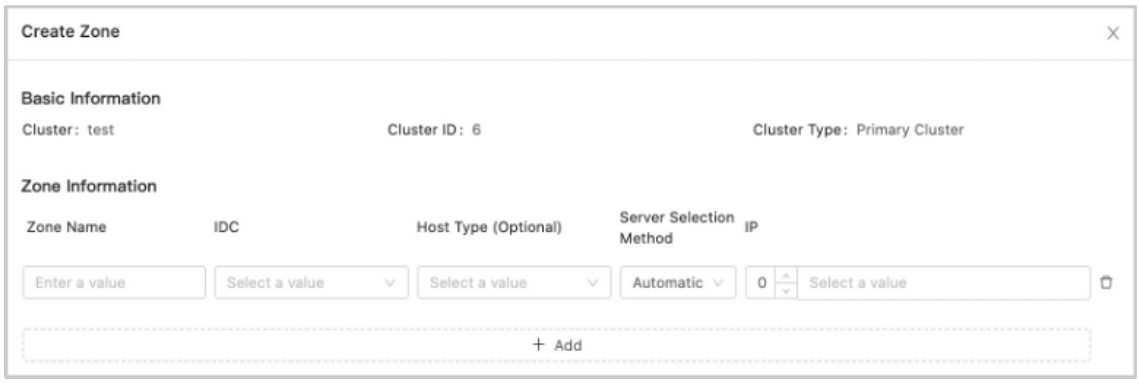
Create a zone

You can create a zone in a cluster that you manage.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the upper-right corner of the **Overview** page, click **Create Zone**.
4. In the panel that appears, specify the zone information.

By default, you can add one zone. To add more than one zone, you can click **Add** in the lower part.



The following table lists the zone information you must specify.

Parameter	Description
Zone Name	Customize the name of the zone.
IDC	The data center to which the zone belongs.
Host Type	(Optional) The IP addresses are filtered based on the host type you select.
Server Selection Method	You can select **Automatic** or **Manual**.
IP	You can select multiple IP addresses. If you select **Automatic** from the **Server Selection Method** drop-down list, you only need to specify the number of IP addresses. OCP automatically selects the specific number of available physical machines. If you select **Manual** from the **Server Selection Method** drop-down list, you must manually select several IP addresses from the IP drop-down list.

5. After you configure the preceding parameters, click **OK**.

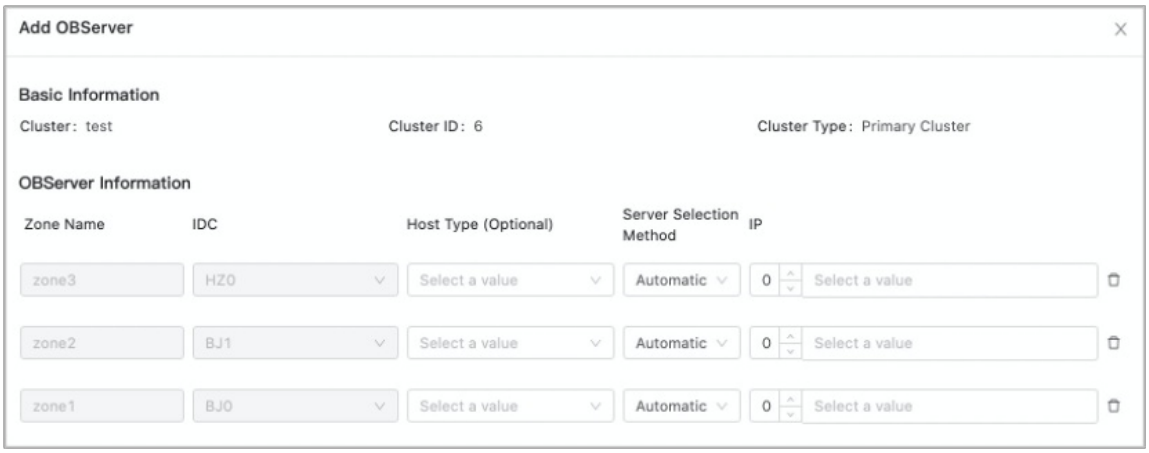
Add an OBServer node

You can add an OBServer node to a cluster that you manage.

Procedure

- Add an OBServer node to a cluster
  i. Log on to OCP. By default, the **Cluster Overview** page appears.
  ii. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
  iii. In the upper-right corner of the **Overview** page, click **Add OBServer**.

iv. In the **OBServerInformation** section, set the host type, server selection mode, and IP address of the OBServer to be added for each zone.



v. Click **OK**.

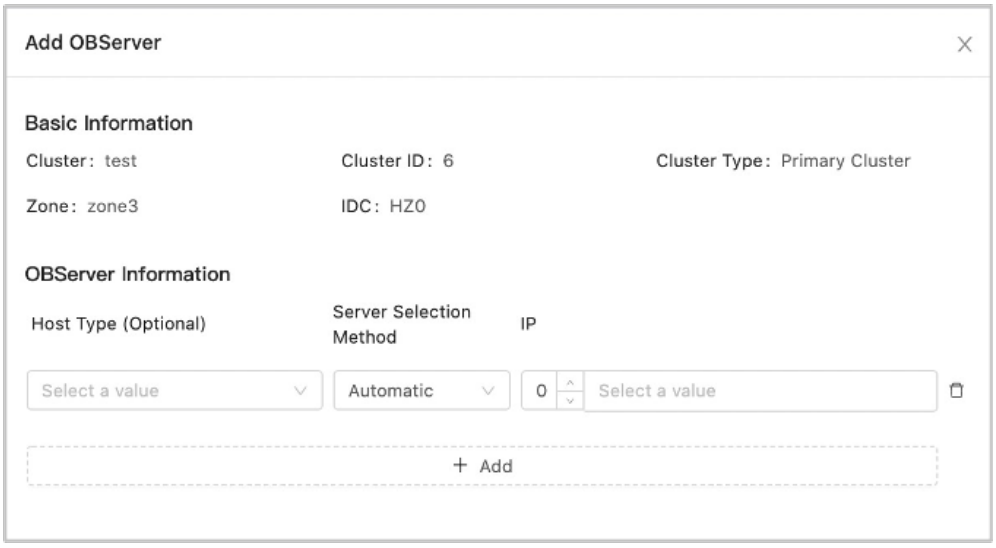
- Add an OBServer node to a specified zone

i. Log on to OCP. By default, the **Cluster Overview** page appears.

ii. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.

iii. In the **Zones** section of the **Overview** page, find the zone to which you want to add an OBServer node, and click **Add OBServer** in the **Actions** column.

iv. By default, one OBServer node is added. To add more OBServer nodes, click the **Add** button in the lower part.



v. Click **OK**.

Restart a zone

You can restart a zone in a cluster. After you restart a zone, the OBServer processes on all nodes in the zone are restarted.

Procedure

1. Log on to OceanBase Cloud Platform (OCP). By default, the **Cluster Overview** page appears.

2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the **Zones** section of the **Overview** page, find the zone that you want to restart, and click **Restart** in the **Actions** column.

   > **Notice**
   >
   > After you restart a zone, the OBServer processes on all nodes in the zone are restarted. Proceed with caution.

4. In the message that appears, click **Restart**.

Stop a zone

You can stop a zone in a cluster. After you stop a zone, the OBServer processes on all nodes in the zone are stopped.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the **Zones** section of the **Overview** page, click the icon for the zone that you want to stop in the **Actions** column and select **Stop**.
4. In the message that appears, click **Stop**.

Delete a zone

You can delete a zone from a cluster. After you delete a zone, all nodes in the zone and node data are deleted.

Procedure

1. Log on to OceanBase Cloud Platform (OCP). By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the **Zones** section of the **Overview** page, find the zone that you want to delete, click the icon in the **Actions** column, and then click **Delete**.
4. In the message that appears, click **Delete**.

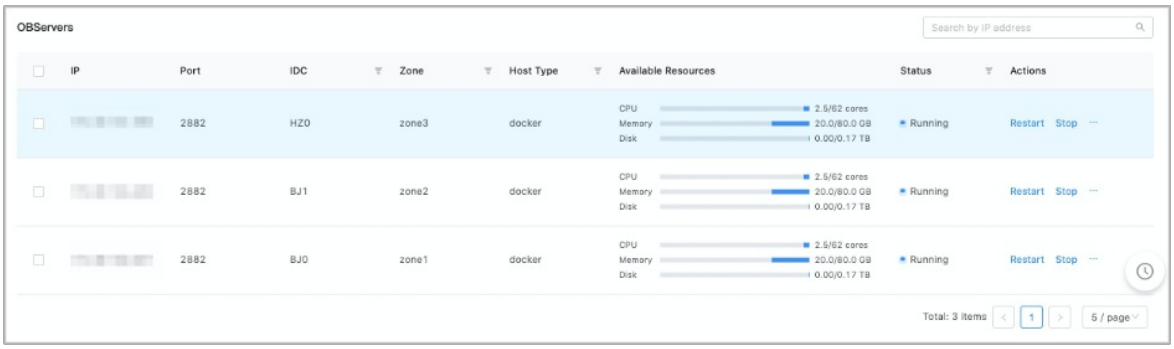
17.1.3.1.3.4. Manage an OBServer process

Restart an OBServer

When an OBServer is faulty, you can restart the OBServer.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to stop.
3. In the **OBServers** section of the **Overview** page, find the OBServer node that you want to stop and click Restart in the **Actions** column.



4. In the **Restart** message, click **Restart**.

Start an OBServer process

If an OBServer process on a node stops, you can start the OBServer process.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the **OBServers** section of the **Overview** page, find the OBServer process that you want to start, and click **Start** in the **Actions** column.
4. In the **Start** message, click **Start**.

Stop an OBServer process

You can stop an OBServer process on a node based on your needs.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the **OBServers** section of the **Overview** page, find the OBServer node that you want to stop and click **Stop** in the **Actions** column.
4. In the **Stop** message, click **Stop**.

Replace an OBServer

You can replace an OBServer on a node based on your needs. After the replacement operation is performed, the system removes the OBServer on the current node from the cluster. Then, the OBServer software is installed on the new node and added to the cluster.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to replace.

3. In the **OBServers** section of the **Overview** page, find the OBServer node that you want to replace. Then, click the icon in the **Actions** column and click **Replace**.

4. In the **ReplaceOBServer** dialog box, select the OBServer node that you want to replace, and click **Replace**.

Delete an OBServer process

You can delete an OBServer process from a node based on your needs.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the **OBServers** section of the **Overview** page, find the OBServer node that you want to delete, click the icon in the **Actions** column, and then click **Delete**.

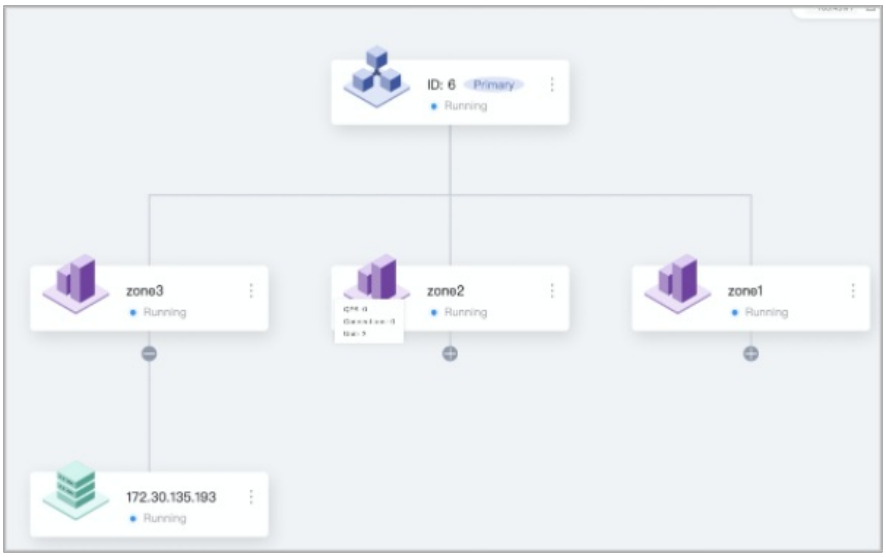
> **Notice**
>
> After you delete an OBServer node, all data on the node is deleted and cannot be restored. Proceed with caution.

4. In the **DeleteOBServer** message, select the OBServer node that you want to delete, and click **Delete**.

17.1.3.1.3.5. View a cluster topology

OCP enables the cluster topology feature. You can view the logical relationships between the zones and OBServer nodes of the cluster and tenant.

In the Clusters section of the Cluster Overview page, click the name of the cluster that you want to manage. In the left-side navigation pane of the page that appears, click Topology. The cluster topology appears.



On the page that appears, you can click the buttons in the upper-right corner to adjust the topology size or refresh the topology.

The topology displays information at three layers: cluster, zone, and server.

Cluster

The cluster layer displays the cluster ID, cluster type, and current status of the cluster. The cluster type can be primary or secondary.

If the cluster is a secondary cluster, you can also view the synchronization latency between the primary and secondary clusters.

The topology also provides an entry to manage the cluster.

For a primary cluster, you can create secondary clusters, add zones or OBServers, upgrade the cluster version, change the password, or restart, stop, or delete the cluster. For a secondary cluster, you can switch it to the primary role or view the cluster.

Zone

The zone layer displays the name and current status of a zone.

You can move the pointer over the zone icon to view the running status of the current zone, such as the queries per second (QPS), number of connections, and number of units.

The topology also provides an entry to manage the zone. In this case, you can add an OBServer and restart, stop, or delete zones.

Server

The server layer displays the IP address and current status of a server.

You can move the pointer over the server icon to view the running status of the current server, such as the QPS, number of connections, and SQL port.

The topology also provides an entry to manage the server. In this case, you can restart, start, stop, replace, or delete servers.

17.1.3.1.3.6. Manage a tenant

View the resource distribution of tenants

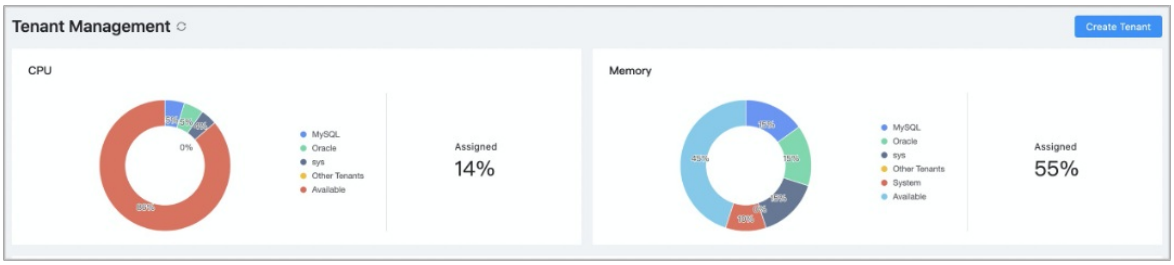
You can view the resource distribution of all tenants in a cluster that you manage.

Prerequisites

All permissions on the specified cluster and permissions to manage the cluster tenants are granted.

Procedure

1. Log on to OceanBase Cloud Platform (OCP). By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the left-side navigation pane of the page that appears, click **Tenant Management**.



4. On the **Tenant Management** page, view information about the resource distribution.

   You can view the percentage of CPU and memory resources used by each tenant and the remaining CPU and memory resources of the system.

   You can move the pointer over a slice in a pie chart to view the resources used by a specified tenant.

Create a tenant

You can create a tenant in the cluster that you manage.

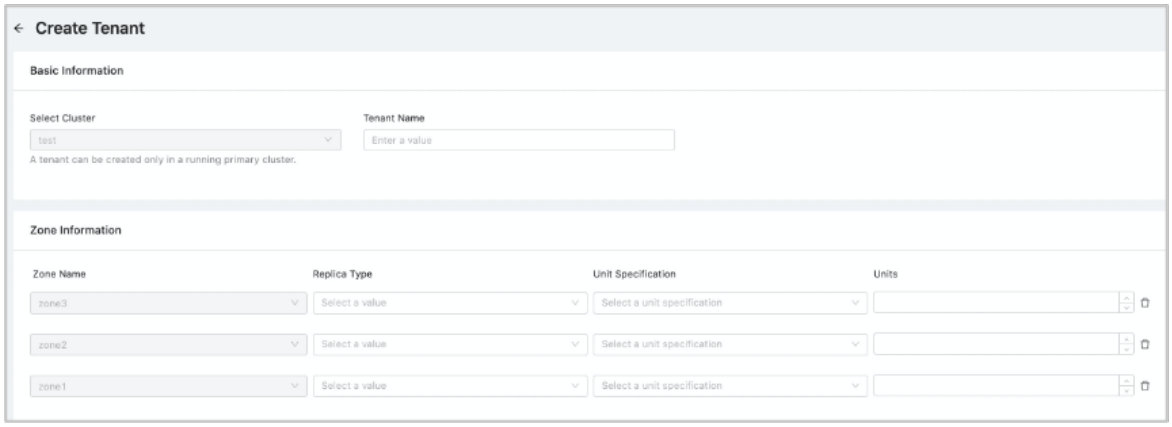
Prerequisites

All permissions on the specified cluster and permissions to manage the cluster tenants are granted.

Procedure

1. Log on to OceanBase Cloud Platform (OCP). By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. On the page that appears, use one of the following two methods to go to the **Create Tenant** page.
   - In the upper-right corner of the **Overview** page, click **Create Tenant**.

- In the left-side navigation pane, click **Tenant Management**. In the upper-right corner of the page that appears, click **Create Tenant**.



4. Enter a name for the new tenant.

   A tenant name must meet the following requirements:

   - The name can contain uppercase and lowercase letters, digits, and underscores (_).
   - The name must be 2 to 64 characters in length.

5. Configure zones for the new cluster tenant.

   By default, the system displays the available zones of the cluster. If you do not want to distribute data replicas to some zones, you can click the Delete icon on the right to delete the zones. The following table describes the parameters for configuring zones.

Parameter	Description
Replica Type	Valid values: ○ Full-featured Replica ○ Read-only Replica ○ Log Replica
Unit Specification	OCP provides a set of unit specifications. For information about how to select a specification, see **OCP resource unit specifications** in the appendix. You can also click **Add Specification** in the lower part of the drop-down list to add a custom specification.
Units	The number of units in the zone. **Note: The number of units cannot exceed the number of OBServers in the zone.**

Parameter	Description
Zone Priority Rankings	The priority of the zone. The setting affects the priority of the primary zones of the system tenant. The list on the left displays all the zones in the current cluster. You can select one or more zones from the left-side list and add them to the right-side list. By default, a zone that you select first has a higher priority than a zone that you select later. If you select multiple zones at a time, they have the same priority. After you move the zones to the right-side list, you can adjust the priority by dragging them. The zone that ranks at the top of the list box has a higher priority than the zones below.

6. Configure basic settings for the new tenant. The following table describes the basic parameters and description.

Parameter	Description
Administrator Password	The password of the tenant administrator. In MySQL mode, the root user is the administrator. In Oracle mode, the SYS user is the administrator. The password must meet the following requirements: ◦ The password can contain 8 to 32 characters in length. ◦ The password must contain at least two digits, two uppercase letters, two lowercase letters, and two special characters. ◦ The password can contain the following special characters: . _+@#$%
Tenant Mode	The MySQL and Oracle tenant modes are supported. The Oracle tenant mode is supported only when the version of the selected ApsaraDB for OceanBase cluster is 2.1 or later.
Character Set and Encoding	In MySQL mode, the following character sets are supported: utf8mb4, binary, gbk, and gb18030. By default, utf8mb4 is used. In Oracle mode, the following character sets are supported: utf8mb4, gbk, and gb18030. By default, utf8mb4 is used.
Remarks	The remarks. This parameter is optional.

Parameter	Description
IP Address Whitelist	You can specify a list of clients that the tenant can connect to. If you leave this parameter empty, the default value is **%**. This indicates that the tenant can connect to all clients. When you configure a custom whitelist, take note that the IP addresses of the OCP server and OBProxy must be added to the whitelist. Otherwise, you cannot manage the tenant on OCP. The whitelist can be in one of the following formats: ○ IP address: 10.10.10.10 and 10.10.10.11 ○ Subnet mask: 10.10.10.0/24 ○ IP address for fuzzy match: 10.10.10.% or 10.10.10._ ○ Combined formats: 10.10.10.10, 10.10.10.11, 10.10.10.%, 10.10.10._, and 10.10.10.0/24

7. Click **Submit**.

View tenants

You can view basic information about all cluster tenants.

Prerequisites

All permissions on the specified cluster and permissions to manage the cluster tenants are granted.

Procedure

1. Log on to OceanBase Cloud Platform (OCP). By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the left-side navigation pane of the page that appears, click **Tenant Management**.
4. In the **Tenants** section of the **Tenant Management** page, you can view basic information about all tenants, including the tenant name, replica type, zone priority, tenant mode, and tenant status.

View tenant monitoring data

You can view the performance of the top five tenants that have the highest workloads in a cluster.

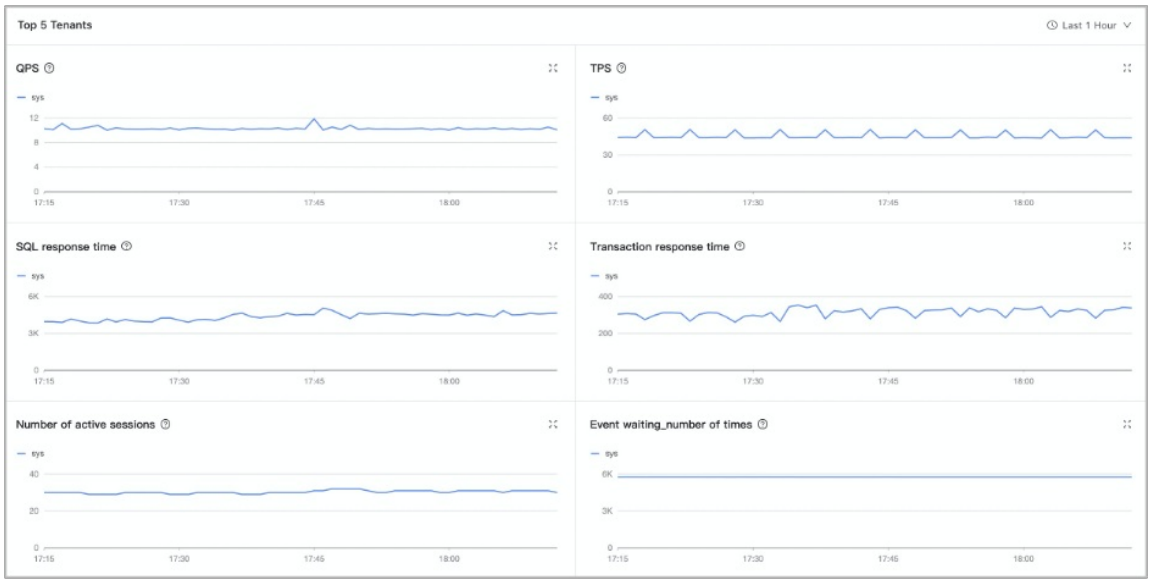
Prerequisites

All permissions on the specified cluster and permissions to manage the cluster tenants are granted.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.

2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the left-side navigation pane of the page that appears, click **Tenant Management**.
4. In the **Top 5 Tenants** section of the **Tenant Management** page, view the performance of the top five tenants that have the highest workloads in the current cluster. The performance monitoring metrics include the TPS, QPS, SQL response time, transaction response time, active sessions, event wait_number, event wait_time, capacity_table quantity, and capacity_partition quantity.
   - You can also view monitoring data of the last hour, last day, or last week based on your needs.



   - The following table describes the performance monitoring metrics.

Metric	Description	Data source
QPS	The average number of SQL statements that are processed per second.	v$sysstat
TPS	The average number of transactions that are processed per second.	v$sysstat
SQL Response Time	The SQL response time. The unit is microsecond.	v$sysstat
Response Time of Transactions	The average amount time for processing each transaction on the server. The unit is microsecond.	v$sysstat
Active Sessions	The number of active sessions.	__all_virtual_processlist

Event Wait_Count	The average number of wait events per second.	v$system_event
Event Wait_Time	The average amount of time spent on waiting for an event.	v$system_event
Capacity_Number of Tables	The number of tables.	gv$table
Capacity_Number of Partitions	The number of partitions.	v$partition

Manage a tenant

You can lock or replicate a tenant.

Prerequisites

You are granted all the permissions for a specified cluster and the permissions for managing tenants in the cluster.

Procedure

- Lock a tenant
  i. Log on to OCP. By default, the **Cluster Overview** page appears.
  ii. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to lock.
  iii. In the left-side navigation pane of the page that appears, click **Tenant Management**.
  iv. In **Tenants** section of the **Tenant Management** page, find the tenant that you want to lock and click **Lock** in the **Actions** column.

  > **Notice**
  > After you lock a tenant, new users cannot connect to the tenant. Proceed with caution.

  v. In the message that appears, click **Lock**.

- Replicate a tenant
  i. Log on to OCP. By default, the **Cluster Overview** page appears.
  ii. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to replicate.
  iii. In the left-side navigation pane of the page that appears, click **Tenant Management**.
  iv. In the **Tenants** section of the **Tenant Management** page, find the tenant you want to replicate and click **Copy** in the **Actions** column.

v. For information about how to specify basic information and zone information on the **Replicate Tenant** page, see Create a tenant.

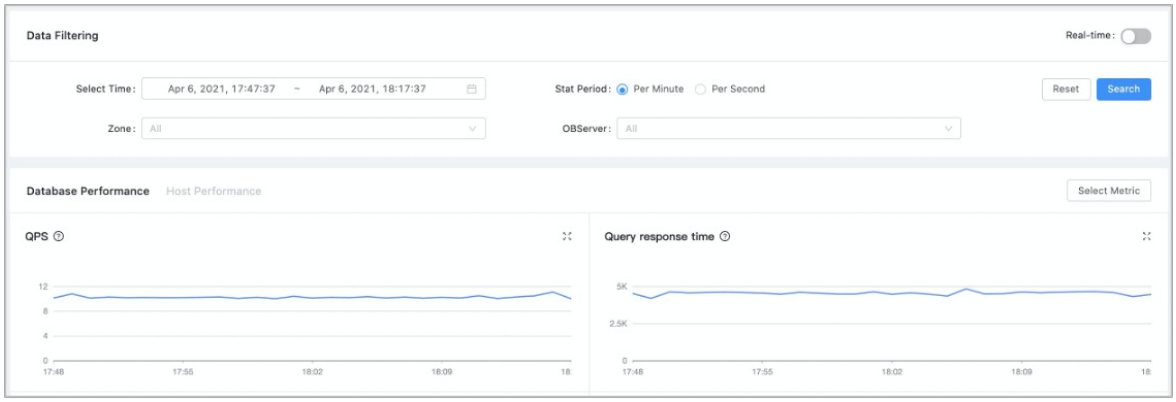
vi. Click **Submit**. After you submit the request, the system creates a new tenant based on the current tenant. However, the data of the current tenant is not replicated.

17.1.3.1.3.7. Performance monitoring

You can view monitoring data of a cluster from different dimensions.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to view.
3. In the left-side navigation pane of the page that appears, click **Performance Monitoring**.
4. In the **Data Filtering** section, specify filter conditions and click **Search**.



The following table describes some of the filter conditions.

Filter condition	Description
Select Time	Select a time range. The monitoring data collected during the specified time range is returned.
Stat Period	Statistical period refers to the statistical period of point data. It allows you to collect data **per minute** or **per second**. That is, you can collect a point per minute or per second. OCP also collects data in another statistical period based on the selected time range. The calculation rules is to collect returned data as close to 1440 points as possible. If the selected time range is too long, the statistical period may longer than 1 minute.
Zone	Select the zone that you want to view.
OBServer	Select the OBServer that you want to view.

5. Switch between the tabs to view the performance metrics of a database or a host in the cluster.

To view other performance metrics, click **Select Metric** on the right of the section. In the Select Metric panel, select the metrics that you want to view. A maximum of 10 metrics can be selected at a time.

For information about database performance and host performance metrics, see Appendix 6. Metrics.

17.1.3.1.3.8. Manage major freezes

Details of major freeze

Major freeze operation allows you to merge dynamic data with static data, but it is time-consuming. If the amount of incremental data generated by minor freezes exceeds a specified threshold, a major freeze operation is triggered. You can view the details about major freeze of clusters.

Procedure

1. Log on to the OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the left-side navigation pane of the page that appears, click **Compaction Management**.
4. On the **Details of Major Freeze** tab, you can view the details of major freeze.

The Details of Major Freeze tab shows the details of the last major freeze in the cluster. If the cluster is in merging state, the major freeze version is displayed. If the state is idle, the information of the last major freeze is displayed.

> **Note**
>
> If the cluster is in merging state, the **Start Time** and **End Time** shows the estimated completion time. The estimated completion time is estimated based on the average value of the last three major freezes and may deviate greatly from the actual time.

In most cases, the estimation is relatively accurate in a stable environment. In a new environment, the accuracy of estimation is low due to a lack of data.

You can also view **Total Partition Replicas** and **Replicas of Partitions After Major Freeze** to learn the progress of a major freeze at the partition level and estimate the completion time.

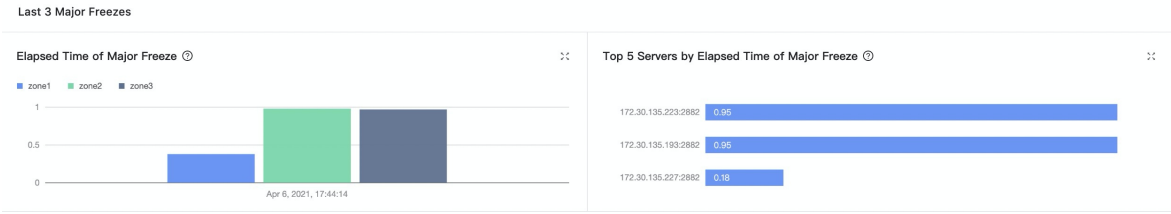
> **Notice**
>
> The completion time estimated directly based on the replicas of partitions after major freeze is inaccurate. The sizes of different partitions vary greatly. The major freeze time of different partitions also varies greatly.

Major freeze statistics

You can view the Major freeze statistics of a cluster.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to view.
3. In the left-side navigation pane of the page that appears, click **Compaction Management**.
4. In the **Last 3 Major Freezes** section of the **Details of Major Freeze** tab, you can view the statistics of major freeze.



Major freeze statistics provide the following four statistical graphs for the last three major freezes:

- Elapsed Time of Major Freeze
- Top 5 Servers by Elapsed Time of Major Freeze
- Top 5 Tenants by Average Minor Freezes
- Top 5 Tenants by Peak Memory Usage Before Major Freeze

The major freeze time and performance data helps you learn the health of the OceanBase cluster because major freeze affects the overall performance of the cluster.

In most cases, we want the daily major freeze is triggered at the time point that we specify.

If the number of times of minor freeze and memory usage exceed a specific threshold, the major freeze is triggered in a cluster. You can determine whether the resources allocated to a tenant are sufficient by viewing the metrics of the tenant. You can upgrade the resource specifications of a tenant to prevent major freeze in a cluster triggered by insufficient resources.

Perform a major freeze

You can trigger an immediate major freeze for an idle cluster. If an error occurs during the major freeze, after you analyze and troubleshoot the error, you can click Continue Major Freeze to remove the error tag and continue the major freeze.

Procedure

1. Log on to OceanBase Cloud Platform (OCP). By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the left-side navigation pane of the page that appears, click **Compaction Management**.

4. In the **Last Major Freeze** section of the **Details of Major Freeze** tab, find **Basic Information**.

5. If **Status** is **Idle**, click **Initiate Major Freeze**.
6. If an error occurs during the cluster major freeze, click Continue Major Freeze to remove the error tag and continue the major freeze.

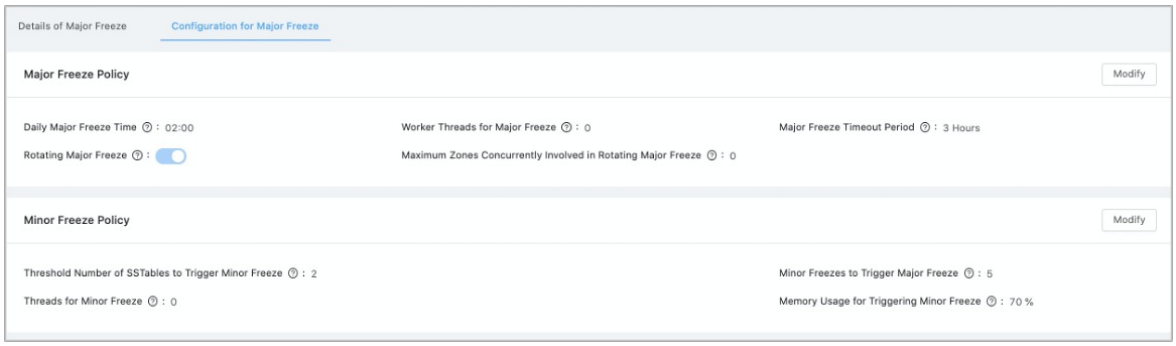
> **Notice**
>
> We recommend that you contact technical personnel for troubleshooting when a major freeze error occurs. If you remove the error tag and continue the major freeze without troubleshooting, the impacts of the issue may be expanded.

Modify major freeze configurations

You can configure major freeze and minor freeze policies based on your needs to perform daily major freeze operations.

Procedure

1. Log on to OceanBase Cloud Platform (OCP). By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the left-side navigation pane of the page that appears, click **Compaction Management**.
4. Click the **Configuration for Major Freeze** tab.



5. Configure the major freeze policy.
   i. In the **Major Freeze Policy** section, click **Modify**.
   ii. Move the pointer over the icon of each configuration item to view system prompts and configure the major freeze policy.
   iii. Click **Save**.

6. Configure the minor freeze policy.
   i. In the **Minor Freeze Policy** section, click **Modify**.
   ii. Move the pointer over the icon of each configuration item to view system prompts and configure the minor freeze policy.
   iii. Click **Save**. Click **Save**.

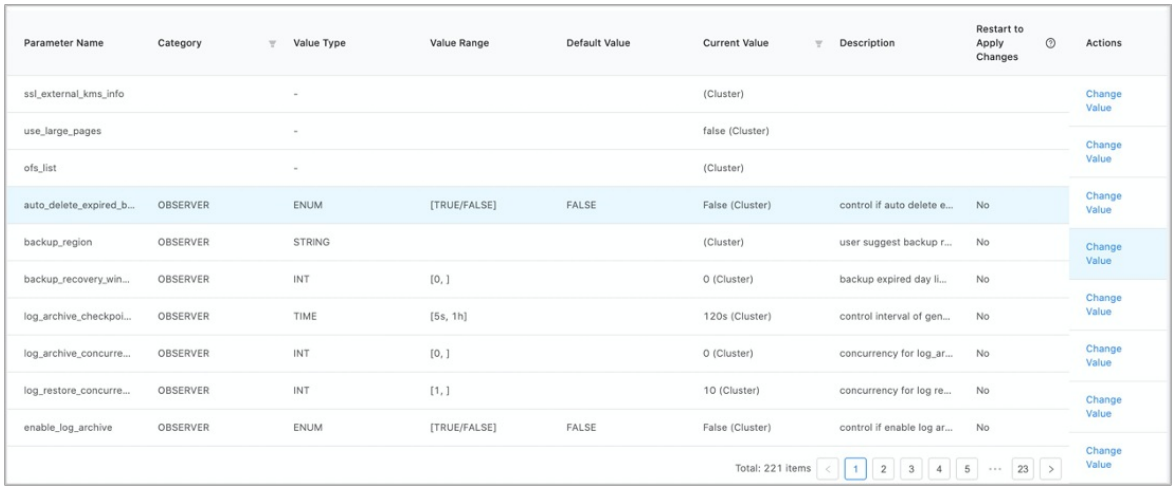
17.1.3.1.3.9. Manage parameters

View parameters

You can view parameters on OceanBase Cloud Platform (OCP). For example, you can view the name, category, value type, value range, default value, current value, and description of each parameter. You can also view whether the parameter takes effect after a restart.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the left-side navigation pane of the page that appears, click **Parameter Management**.
4. On the **Parameters** tab, you can view all parameters in the current cluster. For example, you can view the parameter name, value type, value range, default value, current value, and parameter description. You can also view whether the parameter takes effect after a restart.



Parameter Name	Category	Value Type	Value Range	Default Value	Current Value	Description	Restart to Apply Changes	Actions
ssl_external_kms_info		-			(Cluster)			Change Value
use_large_pages		-			false (Cluster)			Change Value
ofs_list		-			(Cluster)			Change Value
auto_delete_expired_b...	OBSERVER	ENUM	[TRUE/FALSE]	FALSE	False (Cluster)	control if auto delete e...	No	Change Value
backup_region	OBSERVER	STRING			(Cluster)	user suggest backup r...	No	Change Value
backup_recovery_win...	OBSERVER	INT	[0,]		0 (Cluster)	backup expired day li...	No	Change Value
log_archive_checkpoi...	OBSERVER	TIME	[5s, 1h]		120s (Cluster)	control interval of gen...	No	Change Value
log_archive_concurre...	OBSERVER	INT	[0,]		0 (Cluster)	concurrency for log_ar...	No	Change Value
log_restore_concurre...	OBSERVER	INT	[1,]		10 (Cluster)	concurrency for log re...	No	Change Value
enable_log_archive	OBSERVER	ENUM	[TRUE/FALSE]	FALSE	False (Cluster)	control if enable log ar...	No	Change Value

Total: 221 items < 1 2 3 4 5 ··· 23 >

> **Note**
> - **0 (Cluster)** of the **Current Value** parameter indicates all OBServers in the cluster have the same value of this parameter. For example, **backup_region**.
> - If the parameter has different values in the zone or OBServer, a combination of values appears, such as **12;10 Custom**.

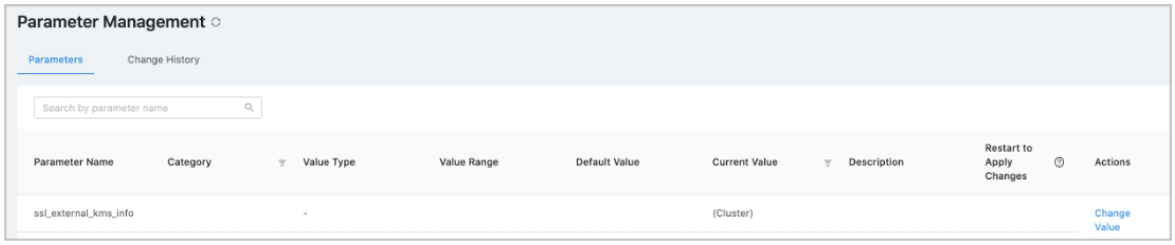
Modify parameter values

You can modify cluster parameter values and the effective scope based on your needs.

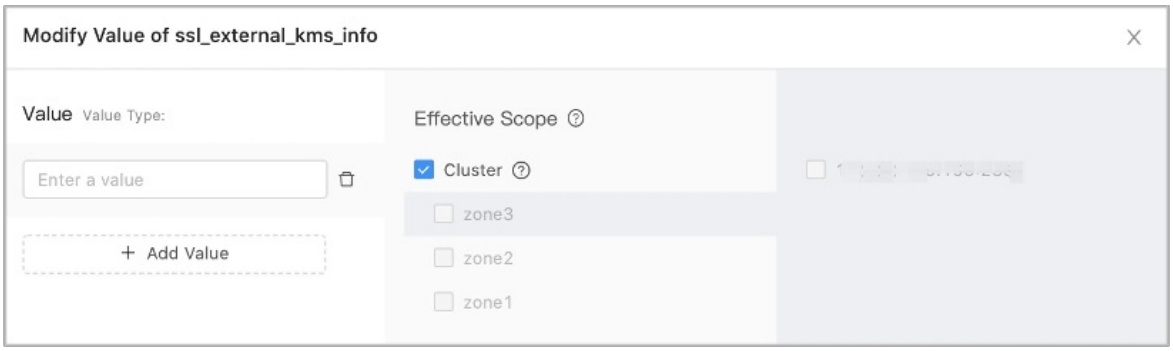
Procedure

1. Log on to OceanBase Cloud Platform (OCP). By default, the **Cluster Overview** page appears.

2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster that you want to manage.
3. In the left-side navigation pane of the page that appears, click **Parameter Management**.
4. Optional. In the upper-left corner of the **Parameters** tab, enter a keyword to search for parameters that meet specified conditions.



5. Find the parameter for which you want to modify the value, and click **Change Value** in the **Actions** column.
6. In the dialog box that appears, modify the parameter value and effective scope, and click **OK**.



You can specify a parameter to take effect in the cluster, zone, or OBServer based on your needs. To configure global settings, select **Cluster** in Effective Scope.

The default effective scope is **Cluster**. If you want to change the effective scope to a zone or an OBServer, you can clear the **Cluster** check box in **Effective Scope**. Then, the system displays a list of zones in the cluster. In this case, if you select a zone, the parameter takes effect in the zone. If you select a zone and then select an OBServer in this zone, the parameter takes effect in the OBServer.

You can also configure a parameter to take effect at different levels. For example, you can set **backup_concurrency** to **10** in **Zone 1**, **12** in **Zone 2**, and 20 in **Server 1**. You can click **Add Value** in the **Value** column to form 3 rows of values. You can specify an effective scope for each value.

> **Note**
>
> After you click **Add Value** and add multiple rows of values, the parameter of the cluster is modified one by one starting from the first row. Every time a modification is completed, a record is generated on the **Change History** tab.

Parameter types

ApsaraDB for OceanBase supports multiple parameter types, including BOOL, CAPACITY, DOUBLE, INT, MOMENT, STRING, STRING_LIST, and TIME.

The following table describes the main parameter types and default units.

Parameter	Description
BOOL	The Boolean type. Valid values: true and false.

Parameter	Description
CAPACITY	The capacity. The unit can be byte, kilobyte, megabyte, gigabyte, terabyte, or petabyte. The unit is not case-sensitive. By default, the unit is megabyte.
DOUBLE	The double-precision floating-point type. A 64-bit value of this type is accurate to 15 digits after the decimal point and has 16 significant digits at most.
INT	The 64-bit integer type. The value can be a positive integer, negative integer, or zero.
MOMENT	The time point. The value is in the hh:mm format, for example, 02:00. You can also set this parameter to disable. This indicates that the time is not specified. MOMENT is used only in the major_freeze_duty_time parameter.
STRING	The string type. You can enter a string based on the parameter description of ApsaraDB for OceanBase.
STRING_LIST	A list of strings. Strings are separated with semicolons (;).
TIME	The time type. The unit can be microsecond, millisecond, second, minute, hour, or day. By default, the unit is second. The unit is not case-sensitive.

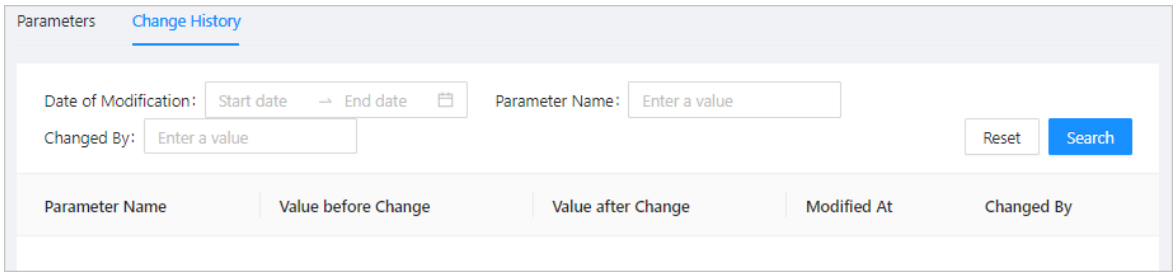
View the history of parameter changes

You can modify parameters in OCP. The Change History tab allows you to view the history of all parameter changes. You can view Parameter Name, Value Before Change, Value After Change, Modified At, and Changed By.

Procedure

1. Log on to OCP. By default, the **Cluster** Overview page appears.
2. In the **Clusters** section of the **Cluster** Overview page, click the name of the cluster that you want to manage.
3. In the left-side navigation pane of the page that appears, click **Parameter Management**.
4. Click the **Change History** tab.
5. You can filter data by specifying **Date of Modification**, **Parameter Name**, or **Changed By**.

   In the **Parameter Name** field, you can enter a keyword to search for parameters that meet specific conditions. However, in the **Changed By** field, you must enter an exact username.



6. In the search result, view the history of parameter changes.

17.1.3.2. Tenant management

17.1.3.2.1. Overview

A tenant is a logical concept. ApsaraDB for OceanBase allocates resources by tenant. You can also manage database objects and resources based on tenants. Tenants play an important role in system O&M, such as the O&M of ApsaraDB for OceanBase clusters. After you log on to OCP, click Tenants in the left-side navigation pane to go to the Tenant Overview page. By default, the Tenant Overview page displays a list of tenants that you have permissions to view and the top five tenants.

17.1.3.2.2. Create a tenant

A tenant is a container for various database objects and resources such as CPU, memory, and I/O. You can create tenants in a cluster based on your business requirements.

You can use one of the following two methods to create a tenant:

- Create a tenant on the Tenant Overview page.
- Create a tenant on the Tenant Management page of a specified cluster.

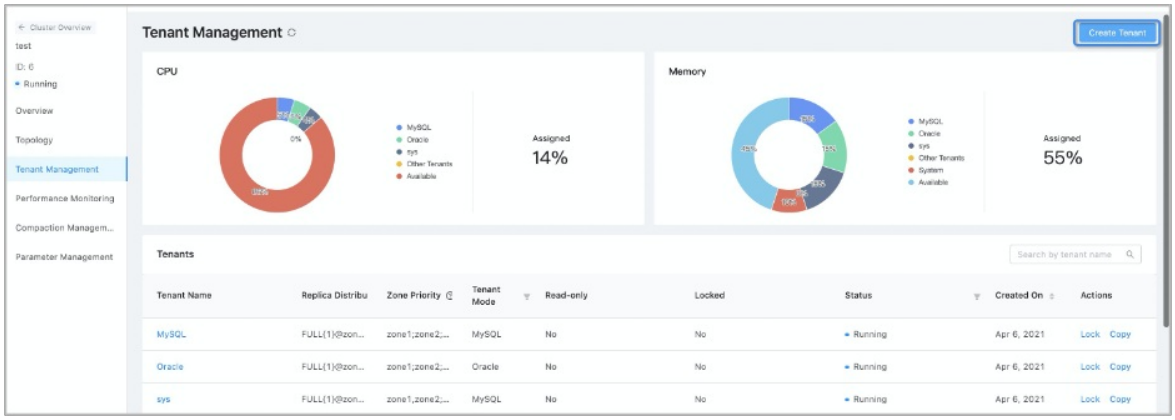
This topic describes how to create a tenant on the Tenant Management page of a specified cluster.

Prerequisites

- The cluster where you create a tenant must be a primary cluster and must be in running state.
- The current user you log on to is the system administrator, OCP tenant administrator role, or other roles that have permissions to manage the cluster.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the **Clusters** section of the **Cluster Overview** page, click the name of the cluster where you want to create a tenant.
3. In the left-side navigation pane of the page that appears, click **Tenant Management**.
4. In the upper-right corner, click **Create Tenant**.

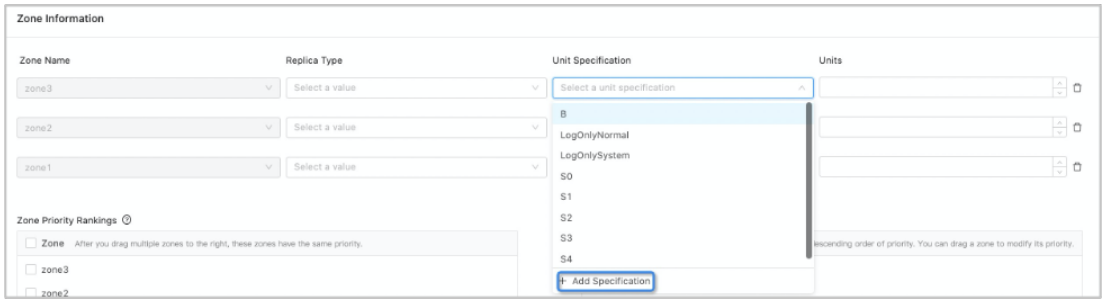


5. Configure **Basic Information**.
   i. The default cluster is the current cluster.
   ii. Enter a **Tenant Name**. The tenant name must be 2 to 64 characters in length and can contain letters, digits, and underscores (_).
6. Configure **Zone Information**.

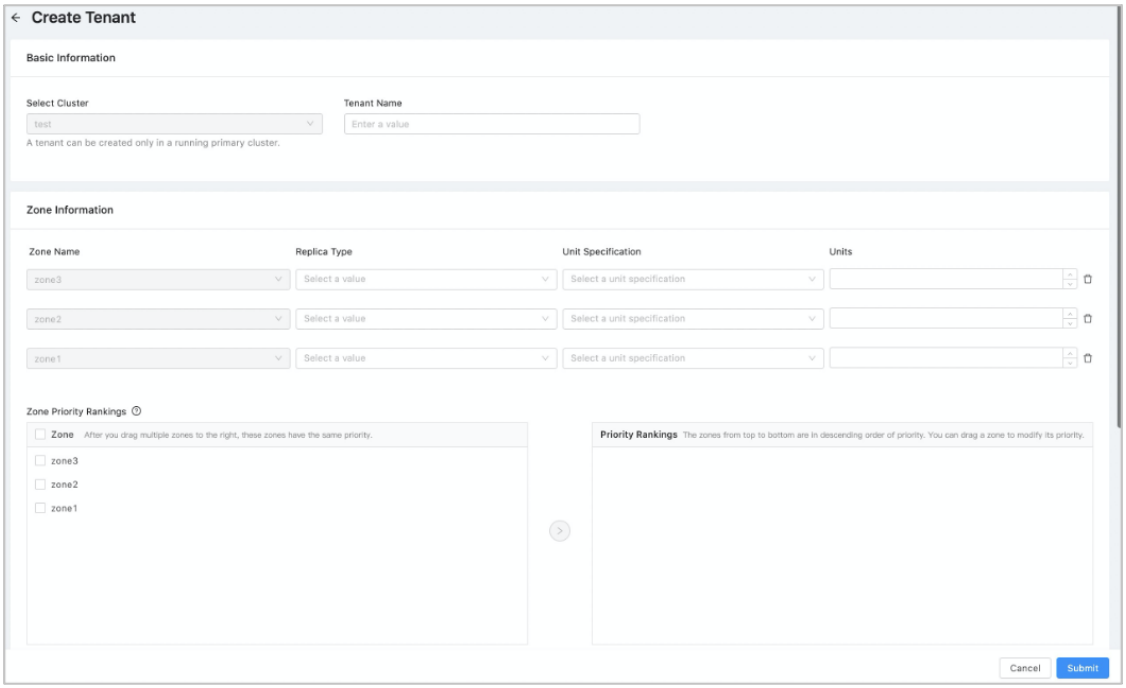
i. Specify the Replica Type, Unit Specification, and Units for zone 1, zone 2, and zone 3.

After you select a cluster, the page shows a list of zones that can be configured based on the zone information of the selected cluster. For zones that do not store data replicas, you can delete the zone by clicking the delete icon on the rightmost side.

- You can select Full-featured Replica, Read-only Replica, and Log Replica from the Replica Type drop-down list.
- OCP provides a set of built-in Unit specifications. You can also click Add Specification at the bottom of the Unit Specification drop-down list to add custom specifications.
- Specify the number of units in the zone. Note that the number of units cannot exceed the number of servers in the zone.



ii. Configure the **Zone Priority Rankings**. If you drag multiple zones to the right, these zones have the same priority.



7. Configure the **Basic Settings**.

i. Settings

- If the tenant mode is MySQL, its administrator account is "root".
- If the tenant mode is Oracle, its administrator account is "SYS". The password must be 8 to 32 characters in length and contain at least two digits, two uppercase letters, two lowercase letters, and two special characters. Special characters include: _+@#$%.

ii. Select a **Tenant Mode**.

- You can select Oracle or MySQL.

- The Oracle tenant mode is supported only when the OceanBase version of the selected cluster is 2.1 or later.

iii. Configure the **Character Set and Encoding**.

- If you select the MySQL tenant mode, the following character sets are available: utf8mb4, binary, gbk, and gb18030. Default value: utf8mb4.
- If you select the Oracle tenant mode, the following character sets are available: utf8mb4, gbk, and gb18030. Default value: utf8mb4.

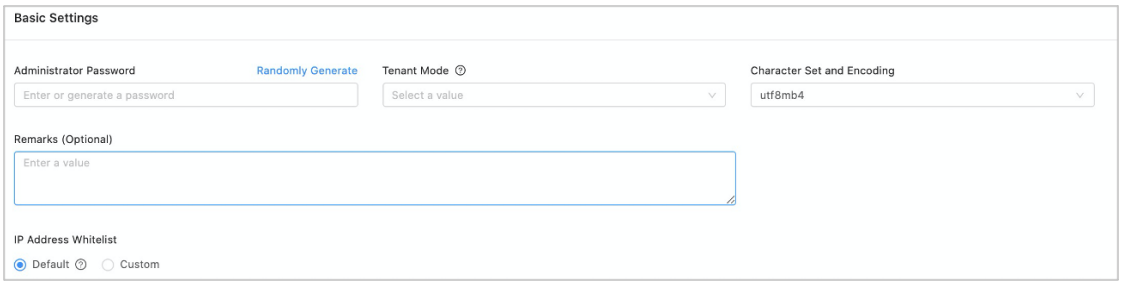
iv. Optional. Configure **Remarks**.

v. Configure the **IP Address Whitelist**.

- You can specify the clients that the tenant can log on to. If you do not specify, the default configuration is "%". It indicates that the tenant can log to all clients. When you customize the whitelist, note that the whitelist must include the IP addresses of the OCP server and the OBProxy on which the OCP server depends. Otherwise, OCP cannot manage this tenant.
- Default: Accesses from all IP addresses is supported.
- Custom: Configure an IP address whitelist. Only accesses from the IP addresses in the whitelist are supported.

- Description of whitelist format:
  - IP addresses. For example, 10.10.10.10,10.10.10.11.
  - Subnet mask. For example, 10.10.10.0/24.
  - Fuzzy match. For example, 10.10.10. % or 10.10.10. _
  - A mix of multiple formats. For example, 10.10.10.10,10.10.10.11,10.10.10. %,10.10.10. _,10.10.10.0/24

> **Note**
> The percent sign (%) indicates that all clients can connect to the tenant.



8. Click **Submit**.

17.1.3.2.3. View tenants

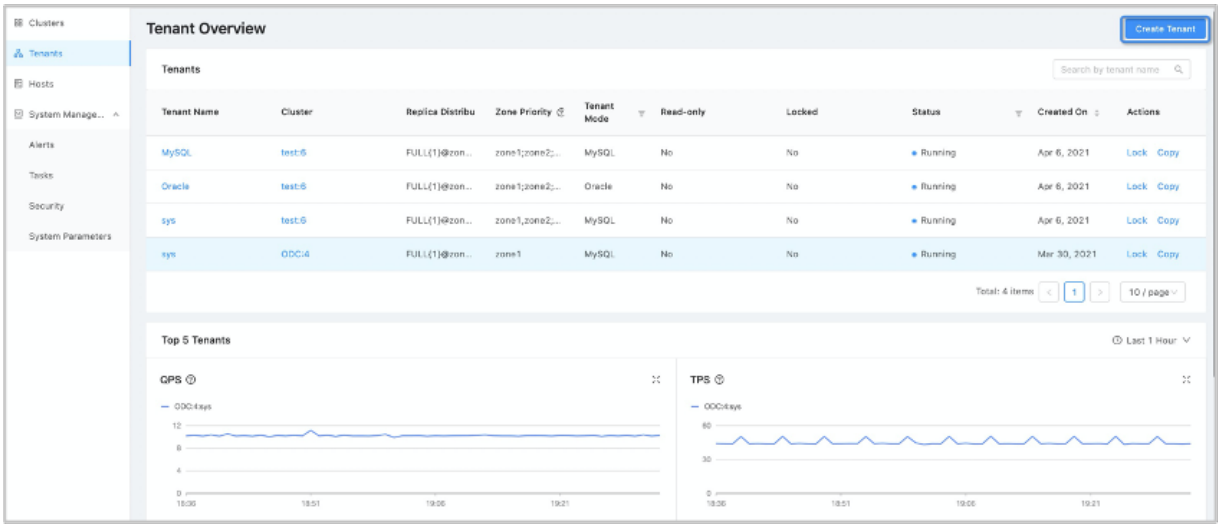
Assume that you have been granted the read-only or all permissions on specified tenants. On the Tenants page, you can view the details and monitoring data of the specified tenants.

Tenants

The Tenants section displays Tenant Name, Cluster, Replica Distribution, Zone Priority, Tenant Mode, Read-only, Locked, Status, and Created On. In the Tenants section, you can lock or replicate a specified tenant, or enter a keyword to search for tenants whose names meet specified conditions. Paged query is also supported.

Replica Distribution specifies the replica type of the tenant in each zone. Zone Priority specifies the priority of the zones that are to be selected as the primary zone. The replica leader is located in the primary zone.

Assume that you have permissions to manage tenants. In this case, you can create, delete, or modify tenants, or have read-only permissions on tenants. The permissions vary based on users. Some users can manage more tenants than other users. Therefore, the number of tenants displayed in the Tenants section varies based on users. Assume that User A has the read-only permission on a specified tenant, but User B has permissions to modify and delete the tenant. In this case, the Actions column displays different user permissions.



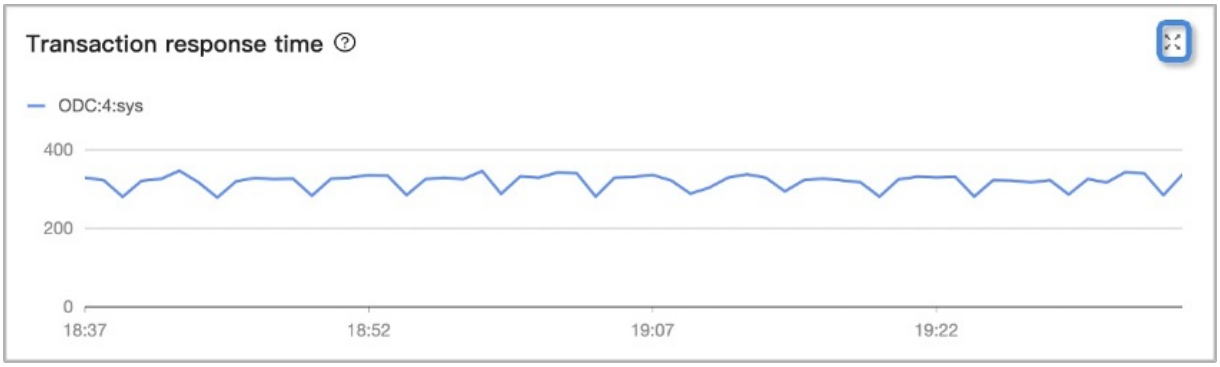
Top 5 Tenants

The **Top 5 Tenants** section displays the performance of the top five tenants in all clusters during a recent period of time. The tenants are ranked by the global core monitoring metrics. The following list describes the core monitoring metrics:

- QPS: The average number of SQL statements processed per second. Data source: v$sysstat.
- TPS: The average number of transactions processed per second. Data source: v$sysstat.
- SQL Response Time: The SQL response time. The unit is microsecond. Data source: v$sysstat.
- Transaction Response Time: The average time for the server to process a transaction. The unit is microsecond. Data source: v$sysstat.
- Active sessions: The number of active sessions. Data source: __all_virtual_processlist.
- Event wait_number: The average number of wait events per second. Data source: v$system_event.
- Event wait_time: The average amount of time spent on waiting for an event. The unit is microsecond. Data source: v$system_event.
- Capacity_table quantity: The number of tables. Data source: gv$table.
- Capacity_partition quantity: The number of partitions. Data source: v$partition.

By default, the Top 5 Tenants section displays the monitoring data of the top five tenants in the last hour. You can also select Last 24 Hours or Last 7 Days from the drop-down list.

When you move the pointer over a specific point in time in the window of a metric, the details of the monitoring data appear, as shown in the following figure. You can view the current time, tenant name, and metric value of the tenant.



To zoom in on the window of a monitoring metric, click the zoom-in icon in the upper-right corner of the window. A resized window appears on the current page. In this case, you can move the pointer over a specific point in time to view the metric value. To return to the Tenant Overview page, click the close button to close the window.

17.1.3.2.4. Manage tenants

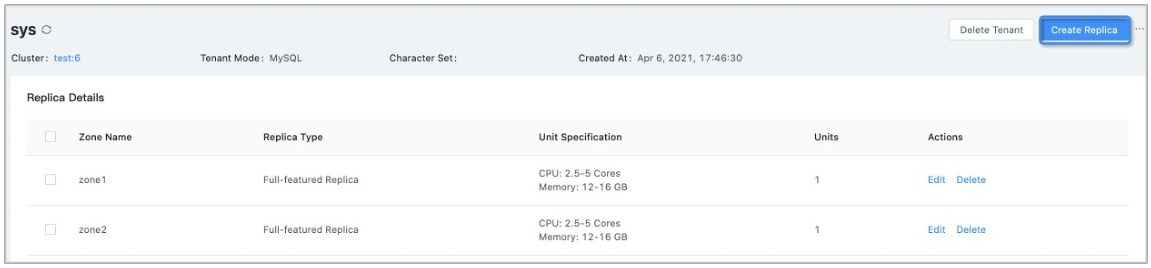
17.1.3.2.4.1. Manage tenant replicas

You can create replicas in all cluster zones under a tenant. You can create one replica in each zone.

Procedure

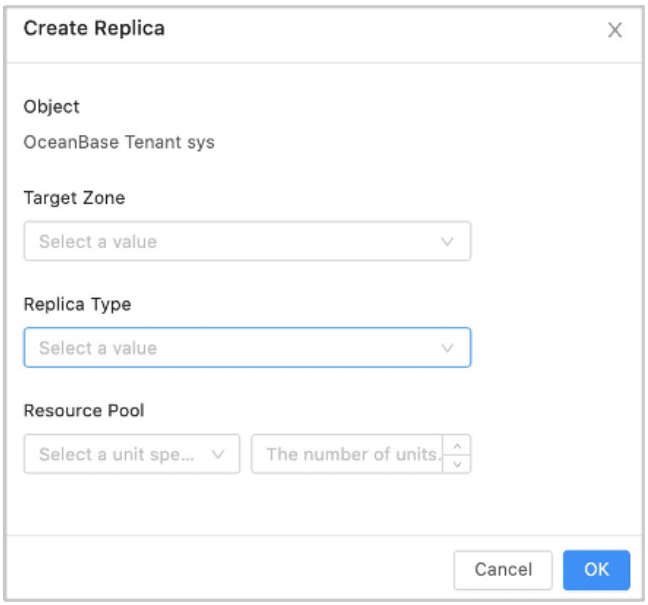
- Create a replica
  i. Log on to OCP. In the left-side navigation pane, click **Tenants**.
  ii. In the **Tenants** section, find the tenant that you want to manage, and click the tenant name.
  iii. In the upper-right corner of the page, click **Create Replica**.

> **Note**
> If replicas are created in all cluster zones under a tenant, you cannot create replicas for the zones.



  iv. Specify **Target Zone**, and add a replica to the zone. You cannot select a zone that has an existing replica.

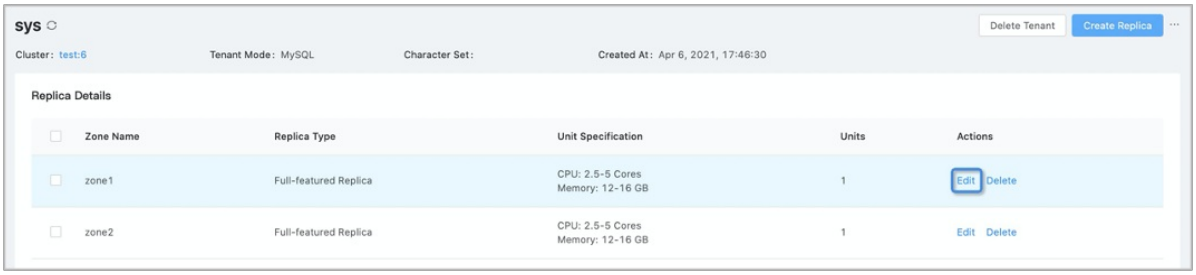
v. Specify **Replica Type** and **Resource Pool**. For more information, see Create a tenant.



vi. Click **OK**.

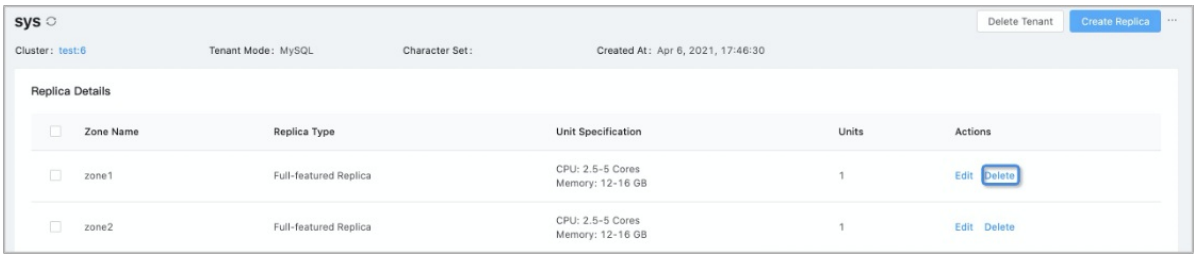
- Edit a replica

  On the tenant overview page, you can view all replicas under the current tenant and the replica details. The Replica Details section displays the distribution of replicas in the zones under the tenant. For example, you can view Zone Name, Replica Type, Unit Specification, and Units. To modify a replica, find the replica that you want to edit, and click **Edit** in the **Actions** column. This allows you to re-configure the replica under the tenant.



- Delete replicas

  You can delete replicas that you no longer need as an administrator. Find the replica that you want to delete, and click Delete in the Actions column. In the message that appears, click Delete to delete the replica from the zone.



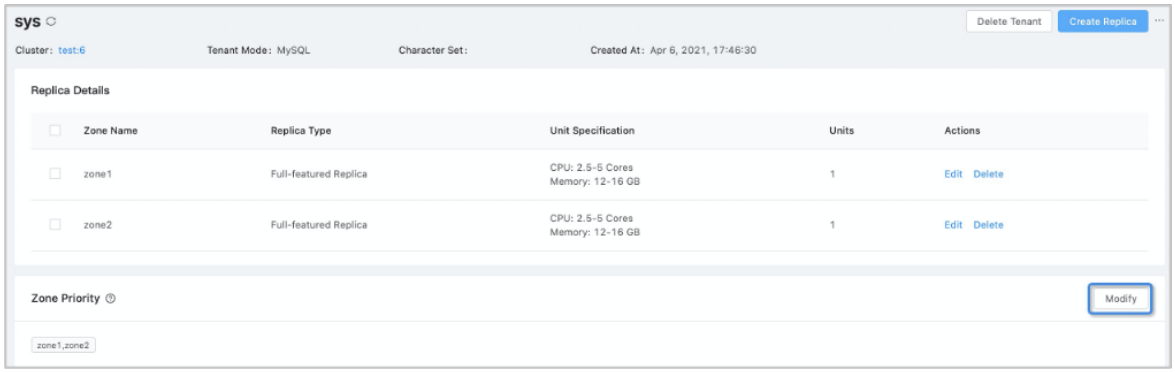
> **Note**
>
> Assume that the system tenant of an ApsaraDB for OceanBase cluster has five replicas in the zones.If two zones are stopped due to exceptions, you cannot delete the replicas. However, you can delete the replicas from the failed zones at a time.

17.1.3.2.4.2. Modify the zone priority

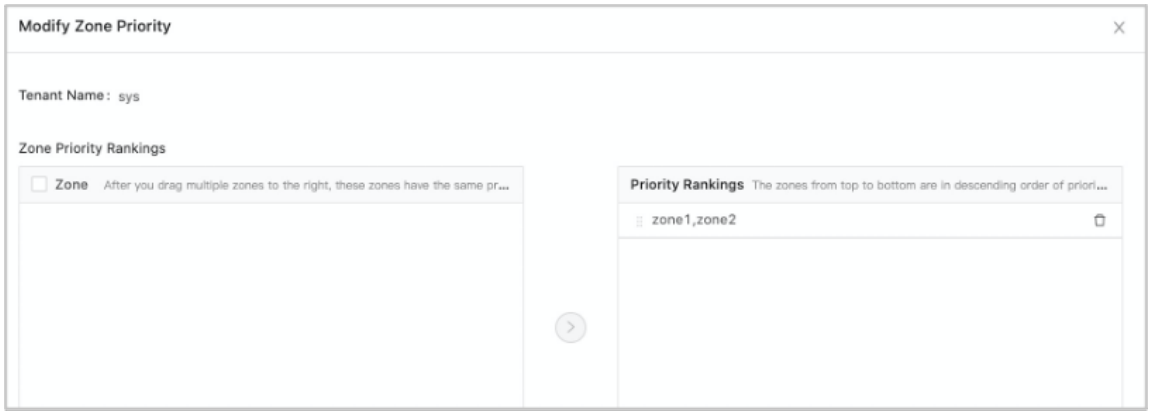
The zone priority specifies the priority of the zones that are to be selected as the primary zone. The replica leader is located in the primary zone. The zone priority is set when you create a tenant. After a tenant is created, you can also modify the zone priority based on your needs.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, click **Tenants**.
3. In the **Tenants** section, find the tenant that you want to manage, and click the tenant name.
4. In the **Zone Priority** section, click **Modify** in the upper-right corner.



5. Reset **Zone Priority Rankings**.
   - You can select multiple zones at a time. After you drag multiple zones to the right at a time, these zones have the same priority.
   - The zones from top to bottom are in descending order of priority. You can drag a zone to modify its priority.



6. Click **OK**.

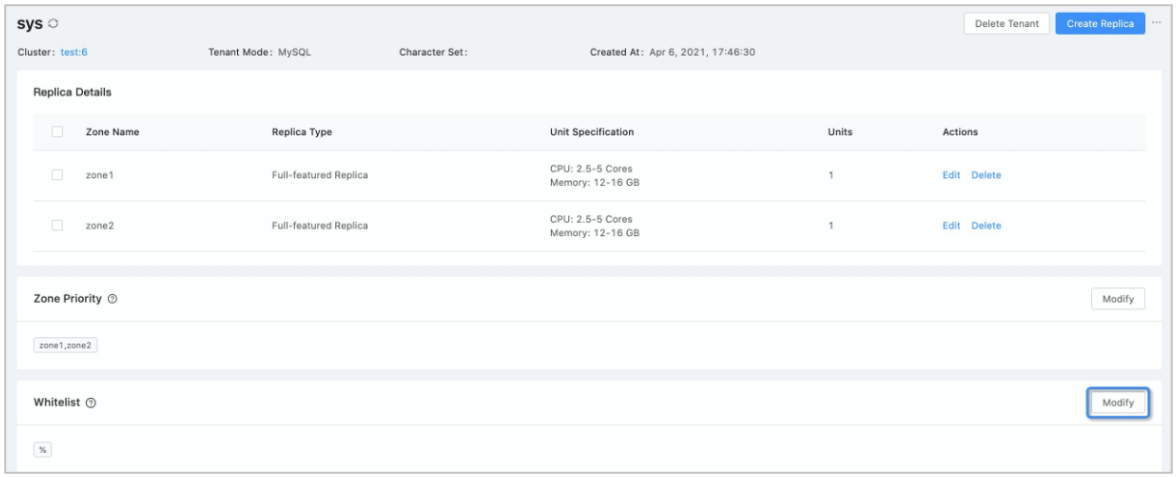
17.1.3.2.4.3. Modify the tenant whitelist

The tenant whitelist is a list of IP addresses that can connect to the tenant. Assume that you do not configure a tenant whitelist when you create a tenant or you want to modify the tenant whitelist. In this case, you can modify the whitelist on the tenant overview page.

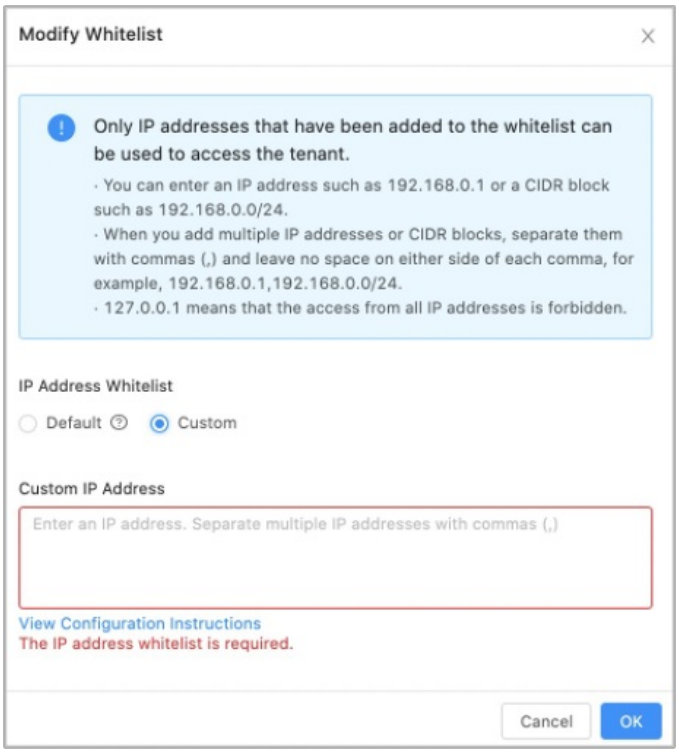
Procedure

1. Log on to OCP.

2. In the left-side navigation pane, click **Tenants**.
3. In the **Tenants** section, find the tenant that you want to manage, and click the tenant name.
4. In the **Whitelist** section, click **Modify** in the upper-right corner.



5. Configure a whitelist.
   - **Default**: All IP addresses can connect to the tenant.
   - **Whitelist**: Only IP addresses in the whitelist can connect to the tenant.Re-configure the whitelist based on the instructions on the page. The whitelist can contain up to 128 characters.



6. Click **OK**.

17.1.3.2.4.4. Change a password

In MySQL mode, the administrator account is the root user. In Oracle mode, the administrator account is the system user. On the Overview page of the specified tenant, you can reset the administrator password for the tenant as a system administrator or tenant administrator.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, click **Tenants**.
3. In the **Tenants** section, find the tenant that you want to manage, and click the tenant name.
4. In the upper-right corner of the page, click the More icon and select **Change Password**.
5. In the Change Password dialog box, enter and confirm the new password.
6. Determine whether to select the **Set the default connection information and save it to the password box** check box.

   We recommend that you select the check box. In this case, when you log on to OCP under the tenant next time, you can use the new password to log on to the cluster.
7. Click **OK**.

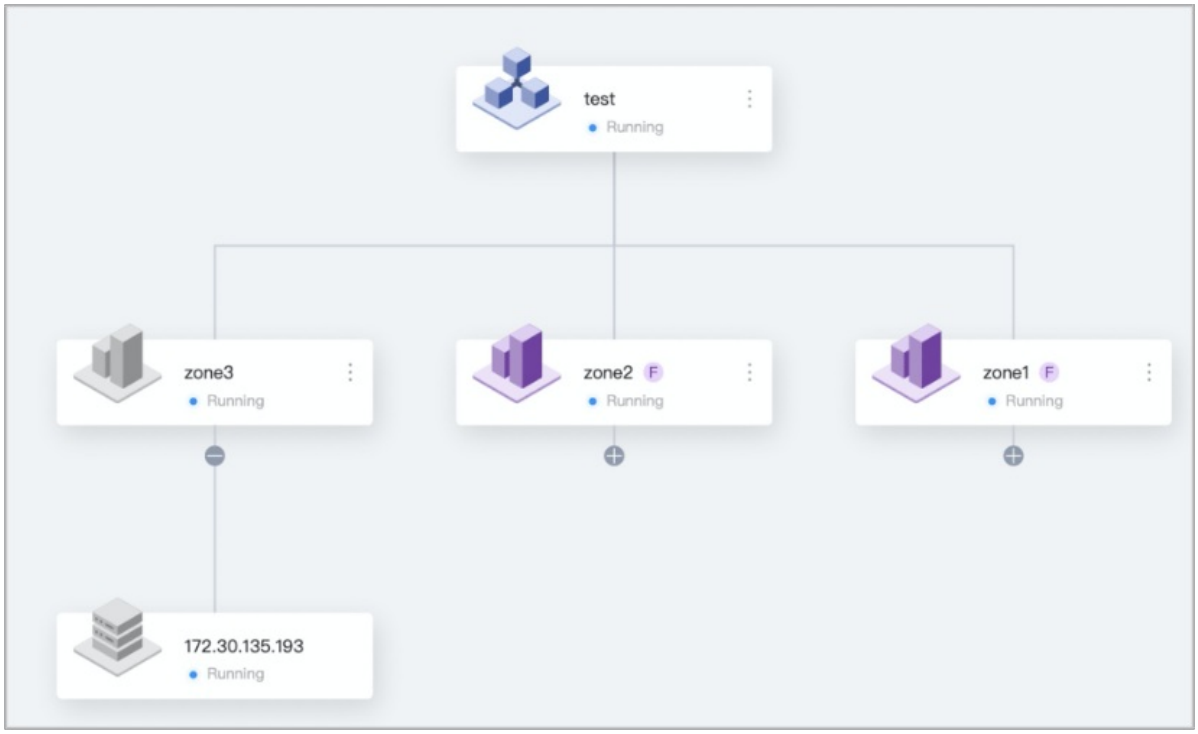
17.1.3.2.4.5. View a tenant topology

After you log on to OCP, click Tenants in the left-side navigation pane to go to the Tenant Overview page. In the left-side navigation pane, click Topology to view the topology of the current tenant.

The topology displays the distribution of replicas in the cluster under the tenant. In the upper-right corner of the page, you can resize or refresh the topology.

The topology displays information at three layers:

- The cluster layer
- The zone layer
- The server layer

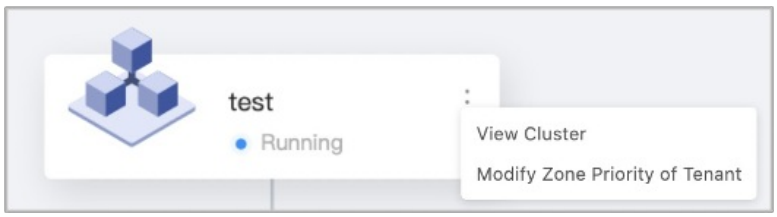


Cluster

The cluster layer displays the name of the cluster under the tenant and the current status of the cluster.

To perform more operations, click the More icon in the upper-right corner.

- To view the cluster topology, click **View Cluster**.
- To modify the zone priority, click **Modify Zone Priority of Tenant**.

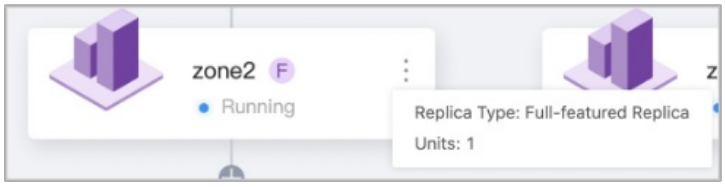


Zone

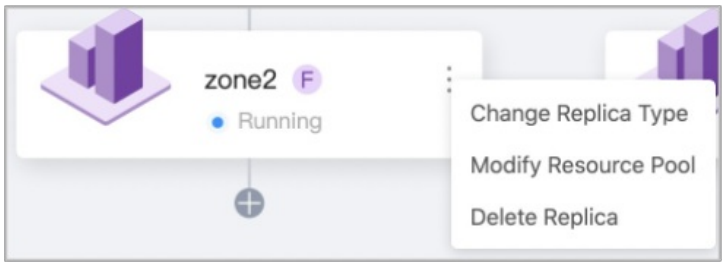
The zone layer displays the name and current status of the zone.

If a zone under the tenant has a replica, the zone is displayed in purple. You can view the abbreviation of the replica type next to the tenant name. If no replica is created in a zone, the zone is displayed in gray.

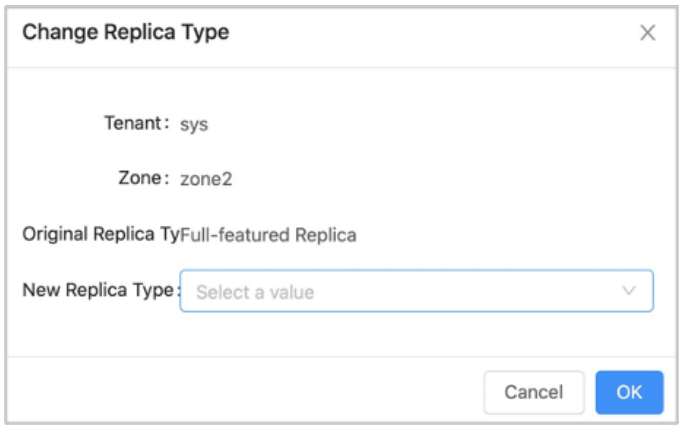
- If you move the pointer over the zone icon, you can view the replica type and number of units in the zone.



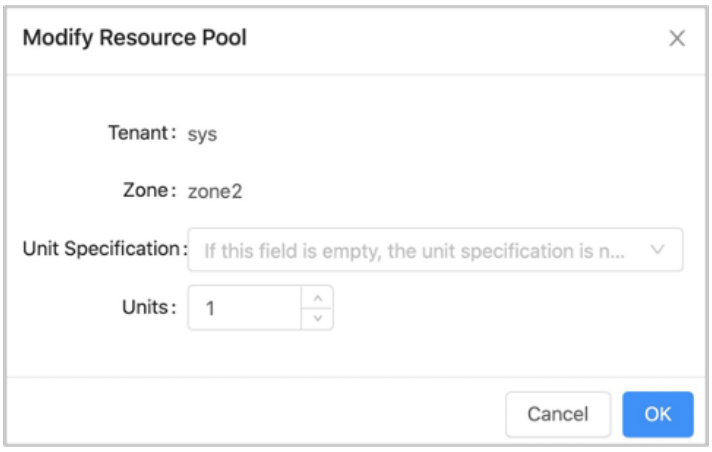
- To perform more operations, click the More icon in the upper-right corner.



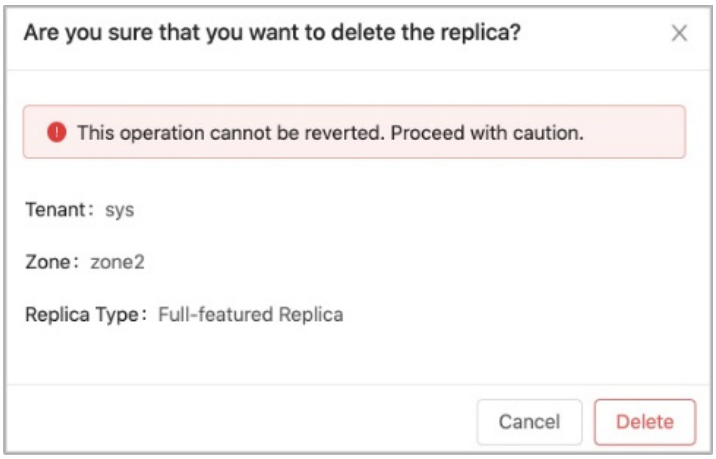
  - To modify the replica type, click **Change Replica Type**. The following dialog box appears.



- To modify the resource pool of a zone under the tenant, click **Modify Resource Pool**. The following dialog box appears.



- To delete a replica, click **Delete Replica**. In the message that appears, click **Delete**.



Server

The server layer displays the IP address, port number, and current status of a server.

If a server under the tenant has a unit, the server icon is green. Otherwise, the server icon is gray.

To perform more operations, click the More icon in the upper-right corner of the server icon.

You can delete a unit. In the message that appears, click **Delete** to delete the specified unit.

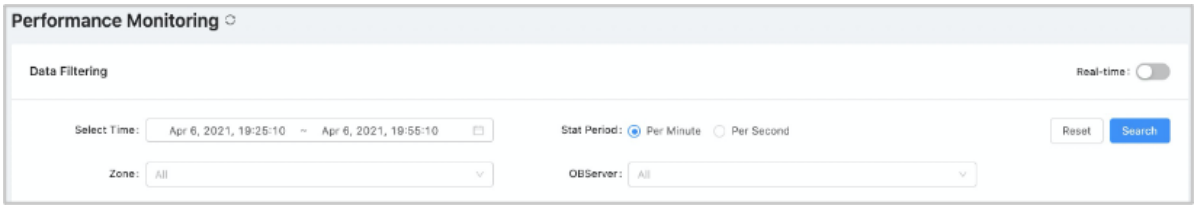
17.1.3.2.4.6. View the performance monitoring data

The Performance Monitoring page displays the performance monitoring data. This topic describes how to filter performance data and view tenant monitoring metrics.

Filter data

You can filter tenant performance data from multiple dimensions based on your needs.

- **Select Time**: You can specify a time range. Data generated during this range will be returned.
- **Stat Period**:
  - The statistical period of each data point. You can select Per Minute or Per Second. Per Minute indicates that a data point is created per minute, and Per Second indicates that a data point is created per second. OCP also calculates a statistical period based on the specified time range.
  - The calculation rule is to return about 1440 data points. If you specify a wide time range, the statistical period may be longer than 1 minute.
- **Zone**: View the performance monitoring data in a specified zone.
- **OBServer**: View the performance monitoring data in a specified OBServer.

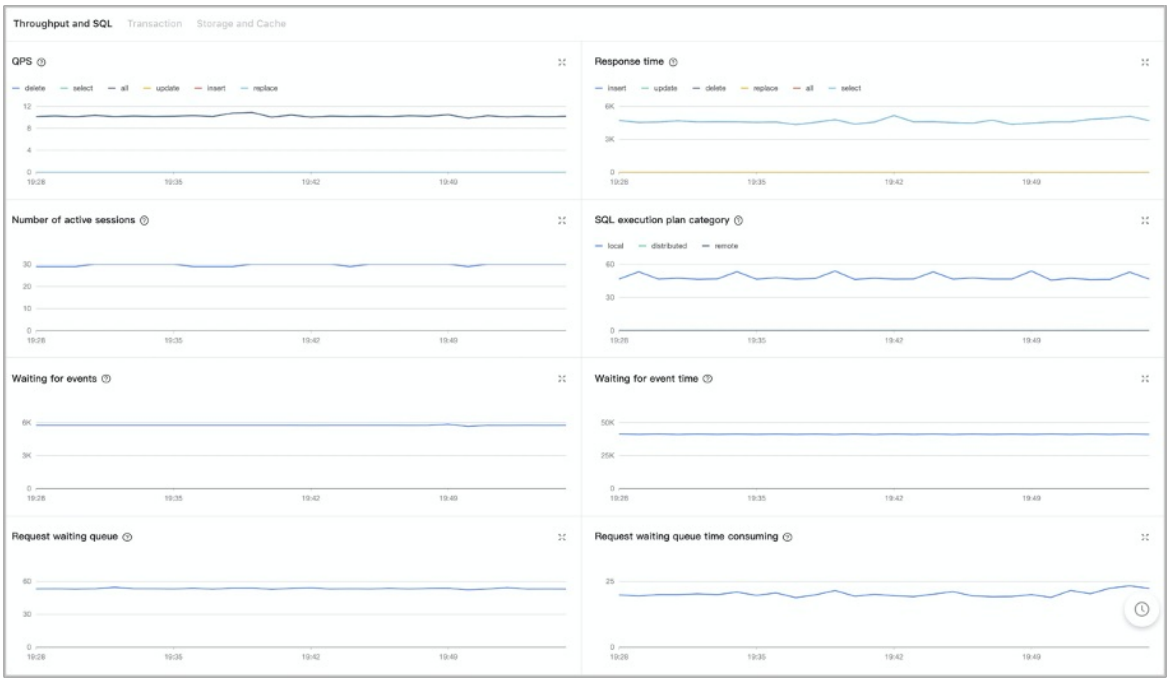


OCP allows you to view tenant performance monitoring data in real time. In the upper-right corner of the page, turn on the **Real-time** switch to view monitoring data in real time. OCP refreshes data on a regular basis based on the specified refresh frequency. By default, OCP displays data that is collected per second in the last two minutes. Refresh Frequency can be 10 Seconds or 1 Second.

Tenant monitoring metrics

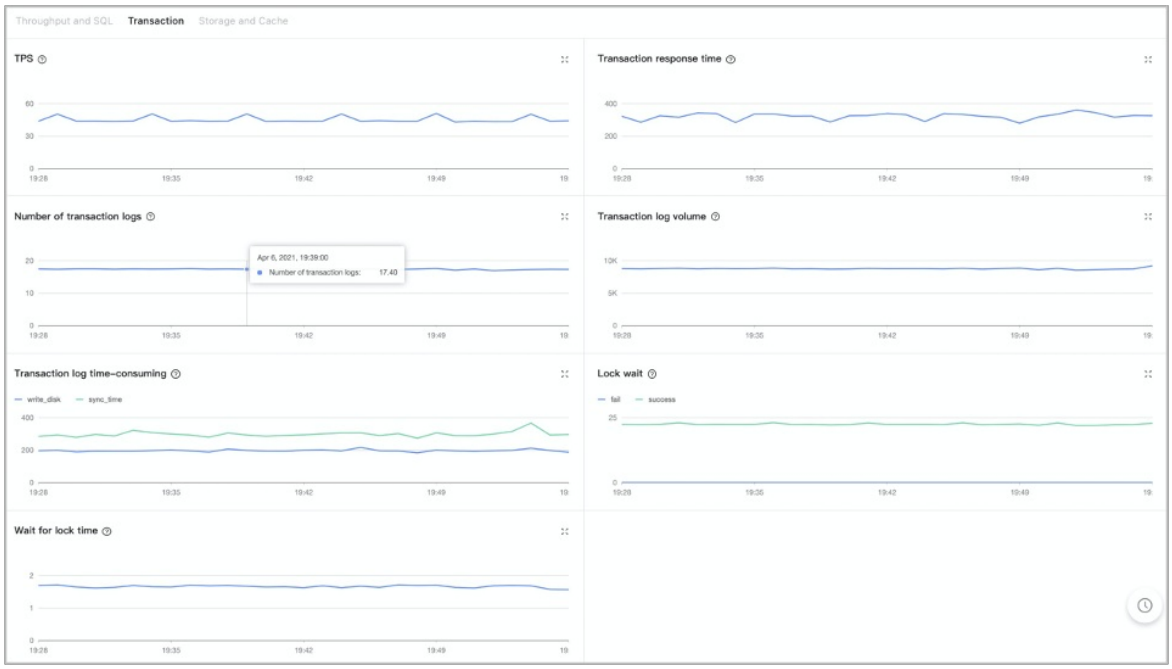
- The Throughput and SQL tab displays the following monitoring metrics:
  - QPS: the average number of SQL statements that are processed per second.
  - Response Time: the response time. The unit is microsecond.
  - Active Sessions: the number of active sessions.
  - Category of SQL Execution Plans: the category of SQL execution plans.
  - Wait Events: the number of wait events per second.
  - Time That Is Consumed by Wait Events: the average time consumed by wait events. The unit is microsecond.
  - Wait Queue of Requests: the number of SQL requests that enter the wait queue per second.

- Time That Is Consumed by the Wait Queue of Requests: the wait time of SQL requests in the wait queue. The unit is microsecond.

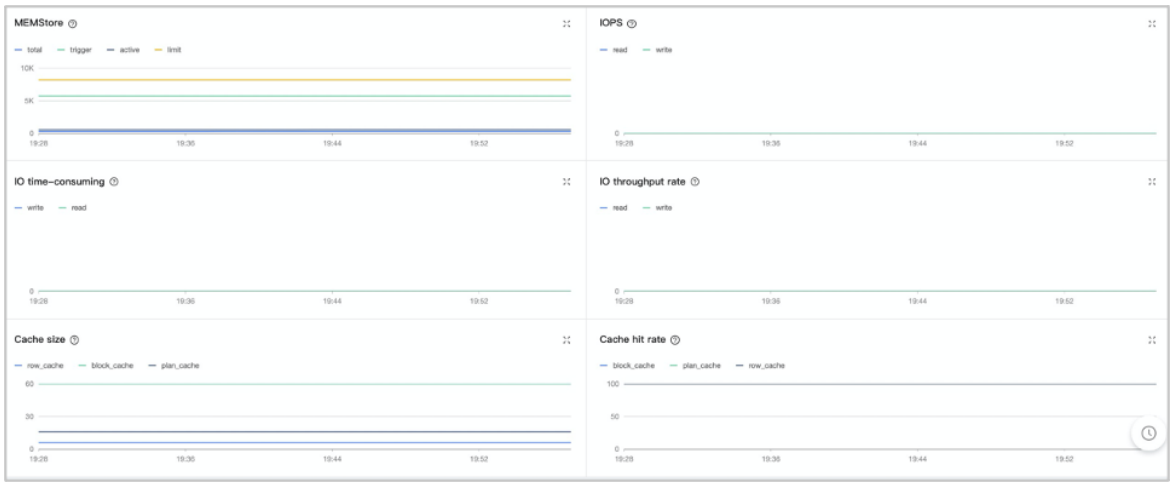


- The Transaction tab displays the following monitoring metrics:
  - TPS: the average number of transactions processed per second.
  - Response Time of Transactions: the average time for processing each transaction on a server. The unit is microsecond.
  - Number of Transaction Logs: the number of transaction logs that are committed per second.
  - Volume of Transaction Logs: the size of transaction logs that are committed per second. The unit is byte.
  - Time Consumption of Transaction Logs: the average time for processing transaction logs on a server. The unit is microsecond.
  - Lock Wait: the number of lock waits for a transaction per second.

- Time That Is Consumed by Waiting for Locks: the average time for lock waits. The unit is microsecond.



- The Storage and Cache tab displays the following monitoring metrics:
  - MEMStore: the size of memory that stores data written into ApsaraDB for OceanBase. The unit is MB.
  - IOPS: the average I/O operations per second.
  - I/O Time: the average time for each I/O operation. The unit is microsecond.
  - I/O Throughput Rate: the average amount of data that is processed by each I/O operation per second. The unit is byte.
  - Cache Size: the cache size. The unit is MB.
  - Cache Hit Rate: the cache hit ratio.



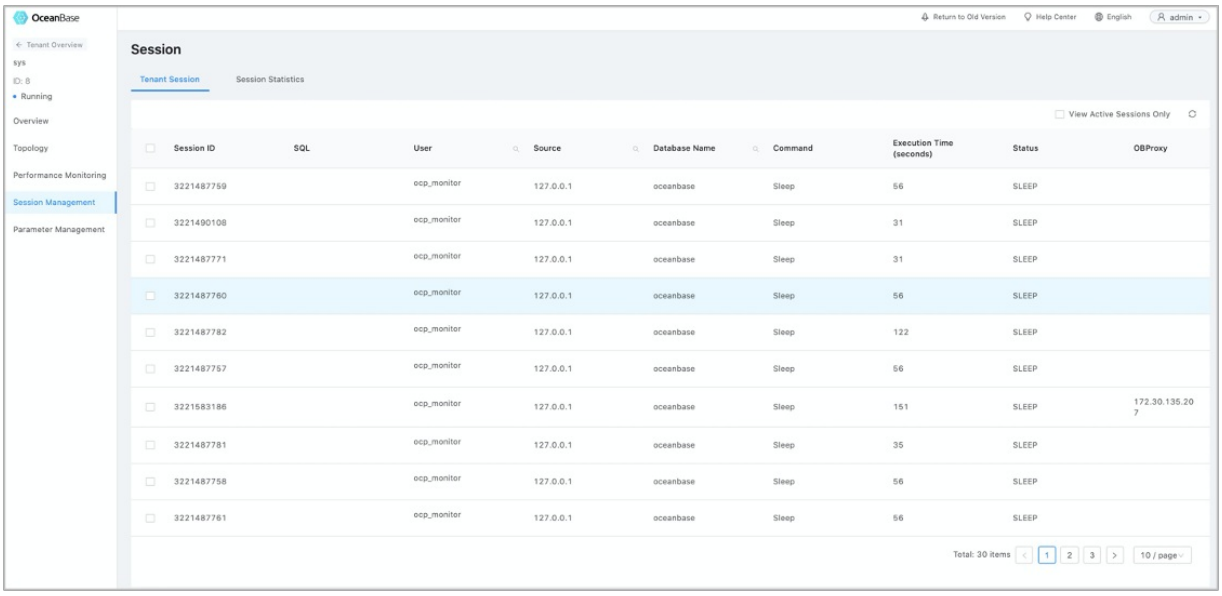
17.1.3.2.4.7. Manage sessions

View tenant sessions

After you log on to OCP, click Tenants in the left-side navigation pane. By default, the Tenant Overview page appears. In the left-side navigation pane, click Session Management. By default, all sessions under the current tenant appear. You can click the refresh button in the upper-right corner to refresh the page.

View sessions

The **Tenant Session** tab displays all current sessions under the tenant. You can view Session ID, SQL, User, Source, Database Name, Command, Execution Time (seconds), Status, and OBProxy.

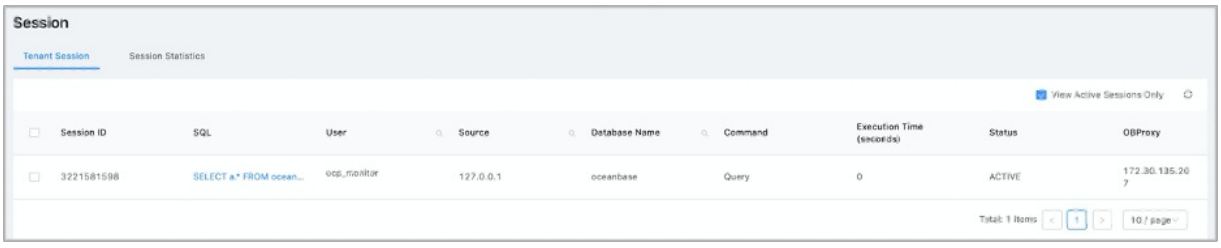


Retrieve sessions

You can retrieve sessions in the User, Source, or Database Name column. To open the search box, click the search icon to the right of each column name.

View active sessions

Assume that you have selected the **View Active Sessions Only** check box in the upper-right corner of the Tenant Session tab. In this case, the sessions are filtered by status. Only sessions in the **ACTIVE** state appear.

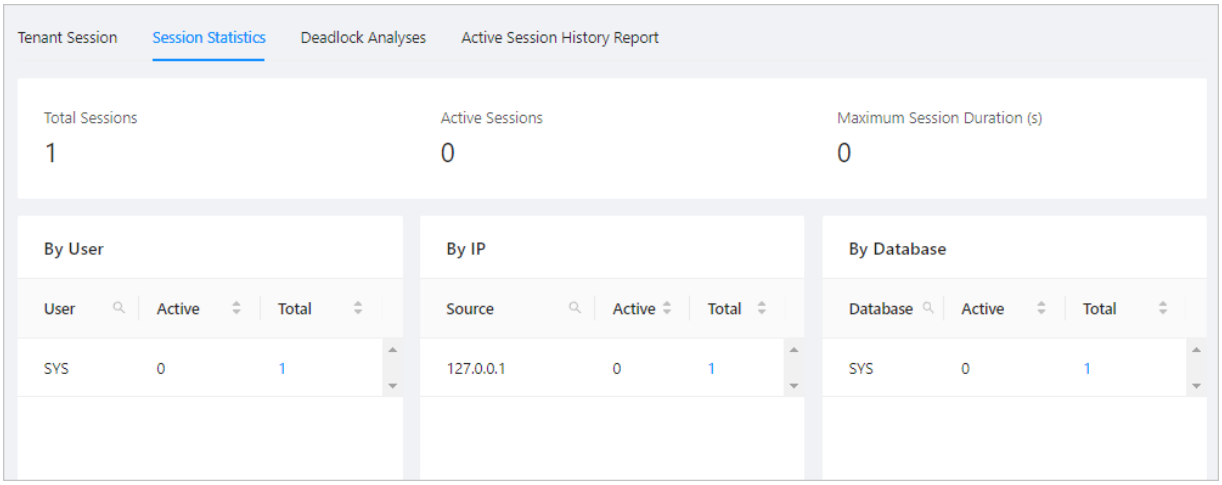


View session statistics

On the left side of the tenants overview page, click Session Management. By default, all sessions of the current tenant is displayed. Click Session Statistics to go to the Session Statistics tab. The statistics of all sessions of the current tenant are displayed on the page.

The statistics include the overall information of sessions, such as the **Total Sessions, Active Sessions, and Maximum Active Session Duration**. You can calculate the **Active Sessions** and **Total Sessions** of a user **By User**, **By IP**, or **By Database**.

> **Note**
>
> The by IP feature is available only in OceanBase 2.2.30 or later.



17.1.3.2.4.8. Parameter management

View the parameters

You can view the name, category, value type, value range, default value, current value, and description of each parameter of the current tenant. You can also view whether the parameter takes effect after the OBServer is restarted. The search box above the list supports fuzzy search by parameter name.

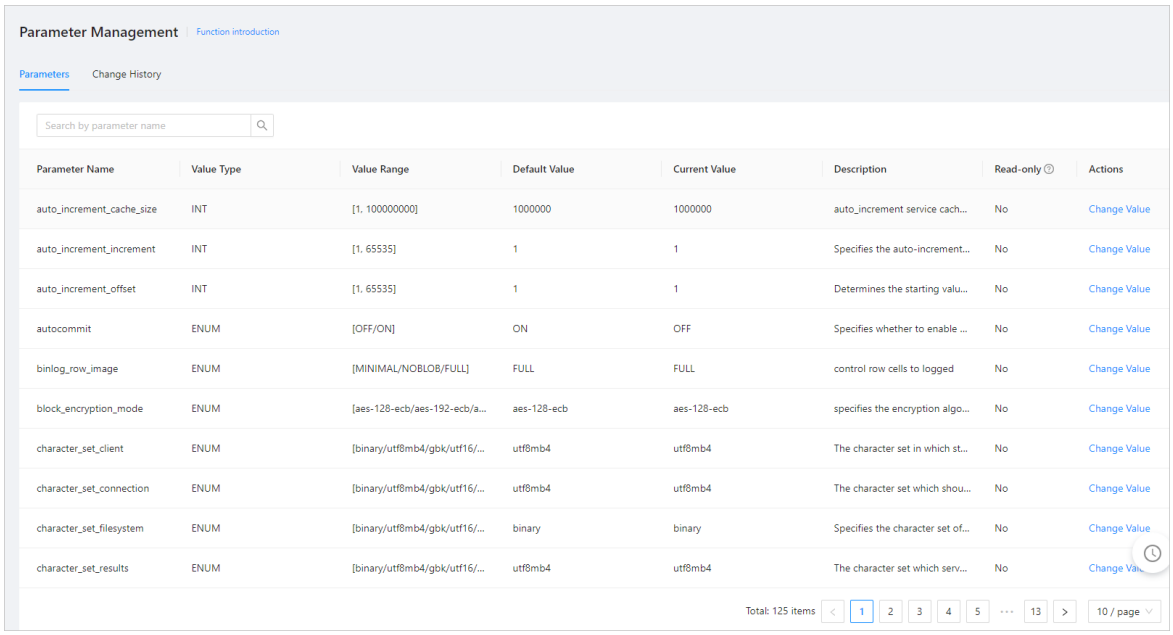
Prerequisites

The password of the root user of the tenant stored in the password box is correct and the tenant whitelist is configured correctly.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, click **Tenants**. In the Tenants, click the name of the tenant that you want to view.
3. In the left-side navigation pane of the page that appears, click **Parameter Management**.
4. On the **Parameters** tab, you can view the information of all parameters in the cluster.
   - The information displayed includes the parameter name, value type, value range, default value, current value, description, and whether the parameters are read-only.

- The **Default Value** indicates the default value of the new tenant. Read-only indicates whether the parameter can be modified. If the parameter is editable, the Change Value button is displayed in the **Actions** column.

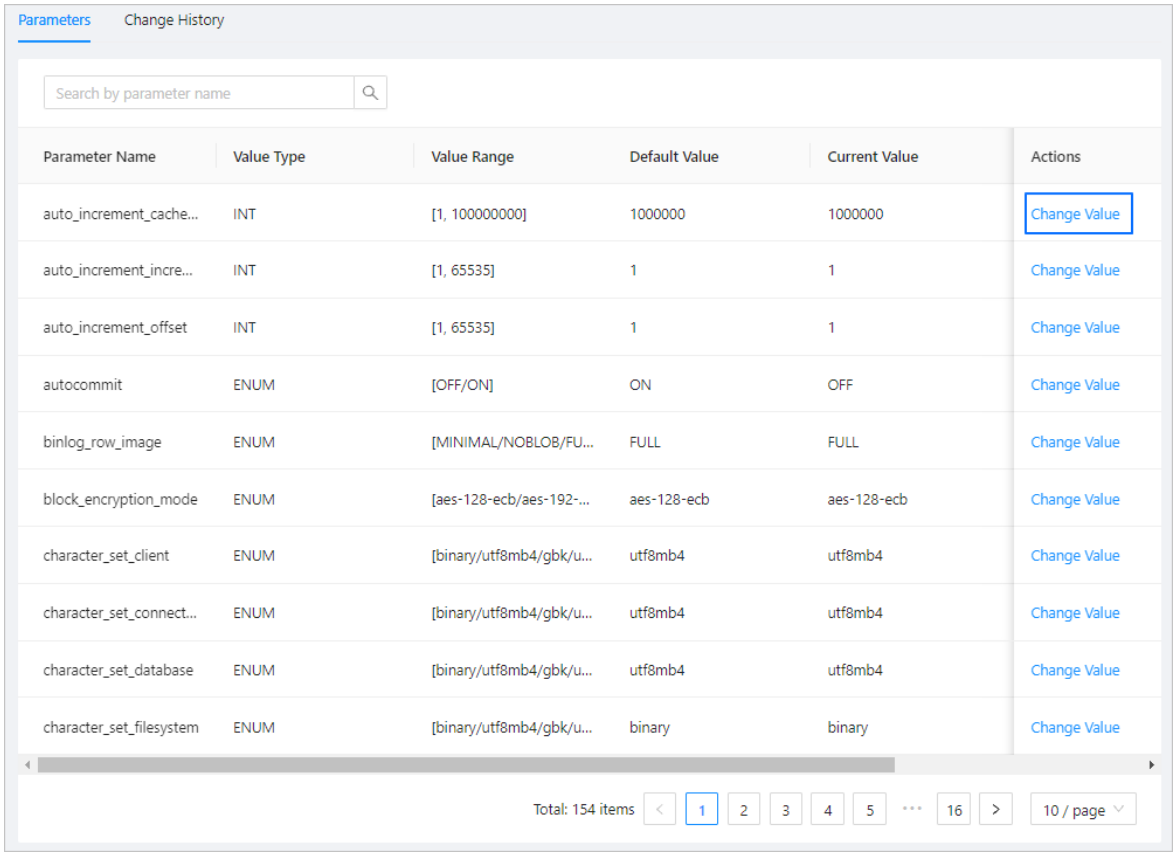


Modify a parameter

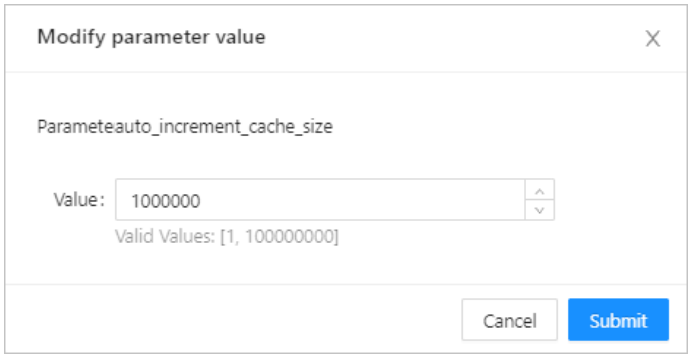
You can modify the tenant parameter values as needed. All changes to the parameter values take effect in real time.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the left-side navigation pane, click **Tenants**.
3. In the **Tenants** section of the **Tenant Overview** page, click the name of the tenant that you want to manage.
4. In the left-side navigation pane of the page that appears, click **Parameter Management**.
5. Optional. In the upper-right corner of the **Parameters** page, enter a keyword in the search box to search for parameters whose names meet specified conditions.
6. Find the parameter you want to modify and click the Modify icon.



7. In the dialog box that appears, modify the parameter value.



8. Click **OK**.

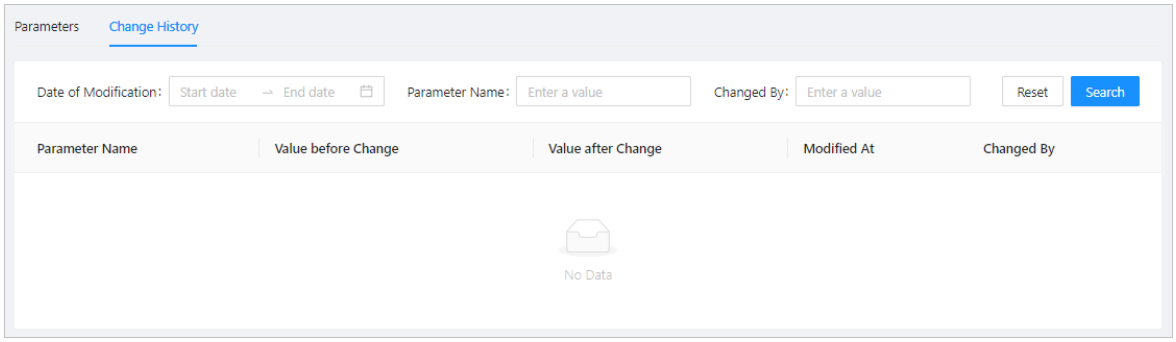
View the parameter modification history

On the Change History tab, you can view the modification records of the parameters of a tenant. The information includes the parameter name, value before and after change of each parameter, the time when a parameter is modified, and the operation user.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the left-side navigation pane, click **Tenants**.
3. In the **Tenants** section of the **Tenant Overview** page, click name of the tenant that you want to view.
4. In the left-side navigation pane of the page that appears, click **Parameter Management**.
5. Click the **Change History** tab.
6. You can filter parameters by specifying **Date of Modification**, **Parameter Name**, or **Changed By**.

The **Parameter Name** field supports fuzzy match. However, the **Changed By** field supports only exact match.



7. In the search results, view the history of parameter changes.

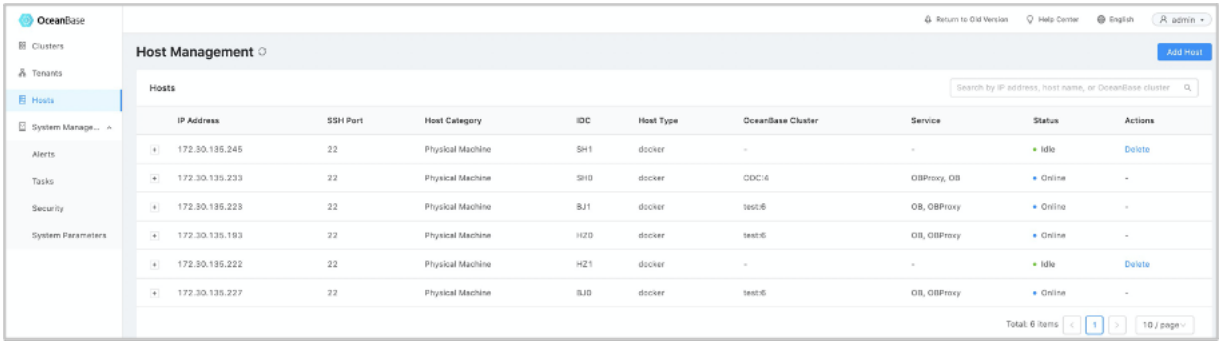
17.1.3.3. Host management

17.1.3.3.1. View host information

Host overview

If you are granted the system administrator or the HOST_MANAGER role, you can click **Hosts** in the left-side navigation pane to go to the Host Management page. The **Hosts** section displays the information about all hosts at the current point in time. You can view IP Address, SSH Port, Host Category, IDC, Host Type, OceanBase Cluster, Service, and Status, as shown in the following figure. You can perform paged queries. You can also search based on host IP addresses, host names, and names of ApsaraDB for OceanBase clusters.

You can expand the collapsed section to view Host ID, Host Name, Operating System, Mapping Port (Host:Container), and Description.



Host status

The host status is an important identifier for monitoring hosts and services running on hosts. OCP supports the following host statuses:

- **Newly Submitted**: The OCP-Agent is to be installed and started on the host.
- **Available**: The host is available for other services.
- **Online**: One or more services are running on the host.
- **Offline**: The host fails to be connected or monitored.
- **Deleting**: The configurations and services on the host are being deleted by OCP.

Host type

The current version supports two types of hosts: physical hosts and container hosts.

- You can deploy multiple services on a physical host. However, to avoid conflicts between services of the same type, you can deploy only one service for each type.
- You can deploy only one service on a container host. Based on the deployment requirements for ApsaraDB for OceanBase clusters, we recommend that you do not deploy ApsaraDB for OceanBase clusters on container hosts.

Port mapping of container ports

OCP servers must initiate remote operations and monitoring to the host irregularly. Container hosts must use the following "port mapping" method based on Docker: Access the IP address and port of the host from the OCP server and map them to the internal port of the Docker container.

By default, four port mappings are needed when you add a container host. The four mappings are created between the ports of processes running in the container and the host ports in the following format:

```
 <Host port>:<Docker port>,<Host port>:<Docker port>. The processes include ocp-agent and node exporter
```

.

The following table describes the mappings.

Service	Container port number	Whether same to the host port	Remarks
OCP-Agent	62888	No	None
OBProxy monitoring	From 62881	No	First available port starting from 62881

17.1.3.3.2. Delete hosts

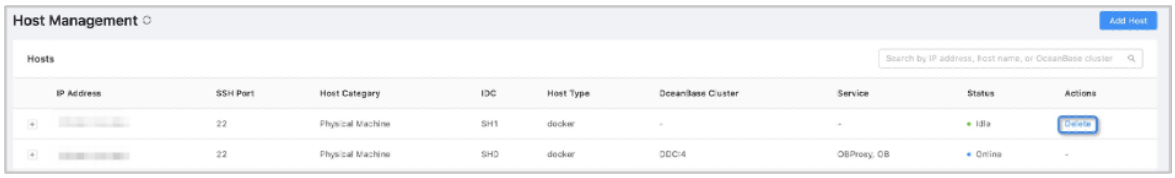
If the Hosts page displays hosts in different states, the shortcut operations Delete and Force Delete are displayed in the Actions column of some hosts in specific states. This indicates that the hosts in the online state can be deleted. The hosts in the newly submitted and offline states can be forcibly deleted.

> Notice
>
> - If you forcibly delete a host, the background tasks related to the host may be interrupted. The services and agent programs running on the host may not be successfully released. Therefore, we recommend that you restore the host to the "online" state before deleting it.
> - We also recommend that you manually check the services and processes running on the host and perform necessary cleanup, such as forcibly stopping processes, removing installation packages, and removing files.

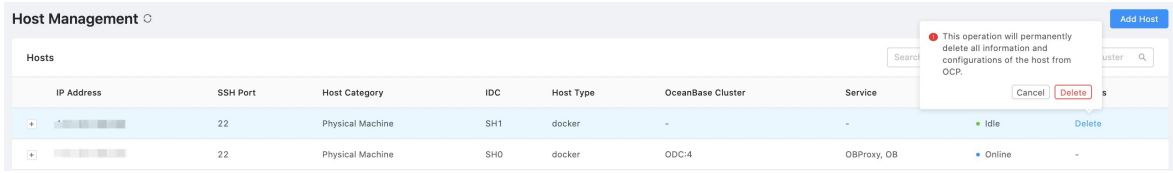
Procedure

1. Log on to OCP.
2. In the left-side navigation pane, click **Hosts**.
3. Find the host that you want to delete forcibly in **Hosts**.

4. In the Actions column, click **Force Delete**.



5. In the message that appears, click **Force Delete**.



17.1.3.4. Alert management

17.1.3.4.1. Overview

This topic describes how OCP enables the alert feature to generate alerts and send notifications when an exception is detected in an ApsaraDB for OceanBase cluster or application.

Core features

- Generate alerts based on alert rules.
- Aggregate alert events to avoid generating a large number of notifications.
- Configure the format and content of alert notifications based on templates.
- Configure alert notification channels. You can configure the HTTP and script channels.

Quick start

Alerts and alert notifications are key features for OCP to monitor service statuses. The users of different roles must perform different operations to use the features.

After OCP is deployed, you must initialize the Alerts module as an administrator. For example, you can add alert items and configure alert channels. You must also manage the existing alert items and channels based on business requirements. OCP allows you to subscribe to alert notifications, and view alert events and notifications for different modules. You can also block alerts.

We recommend that you configure the Alerts module on OCP based on the following process:

1. Perform the following steps as an administrator:
   i. Configure alert notification channels.
   ii. Create alert items.
2. Perform the following steps as a common user:
   i. Subscribe to alerts.
   ii. View alert events.
   iii. Optional. Block alerts.

17.1.3.4.2. Alert related concepts

This topic describes the concepts of alert target, alert scope, alert item, alert, alert item group, alert level, template, alert aggregation group, and alert clearance.

Alert target

An alert target is a target that is monitored by the alert task and uniquely identifies an alert. It can be an OceanBase cluster, a server, or a service.

Based on the alert item, an alert target can be a tag value or a combination of tag values, such as `obregion=obocp:svr_ip=*. *. *. *` identifies a server in the OceanBase Cloud Platform (OCP) cluster.

Alert scope

The alert scope defines the scope of an alert and is consistent with the metric scope. For example, when the CPU utilization exceeds the threshold, it can be a problem for the entire cluster, the tenant, or a single server.

Valid values of the alert scope:

- Cluster
- Tenant
- Application Cluster
- Service
- Server
- Process (reserved)

Alert item

An alert item is the metadata of an alert. It consists of many elements such as the alert type, name, trigger rule, overview, and details.

Alert items can be divided into two types based on how they are generated:

- **Expression-triggered alert items**: Alert items that are created on the console and generated by the alert rule engine based on the monitoring metrics.
- **Custom-triggered alert items**: Alert items that are automatically triggered by other components.

For expression-triggered alert items, the alert rule expressions are configured in the alert items. For custom-triggered alert items, the alert rule expressions are empty.

Alert

An alert is a notification generated by the system when an alert item occurs on a notification target.

For example, when the alarm_b alert item occurs on Server A (an alert target), the alert signal is sent every minute. However, it is counted as one alert on OCP, and only one record is displayed on the Alert Events page.

Alert item group

You can set multiple groups for each alert item for easy configuration of alert subscriptions.

Alert level

Each alert item has an alert level.

Level	Meaning	Color	Description

Level	Meaning	Color	Description
1	Stopped	Purple	The system is completely unavailable and needs immediate recovery. For example, `OBService cannot be started.`
2	Critical	Red	The system availability decreases and the necessary measures must be taken to prevent the system from becoming completely unavailable. For example, `the server memory usage exceeds the threshold of 90% and this condition has lasted for 3 minutes.`
3	Warning	Orange	The system is still available but it is about to become unavailable. You must take measures to prevent the reduction of availability. For example, `the proportion of connections of an OceanBase tenant exceeds the threshold of 80%.`
4	Caution	Blue	Based on the trend, you can tell that the important performance metrics of the system are declining. You can locate potential problems through troubleshooting to prevent the trigger of alerts. This alert level is reserved but no alert matches this level at present.

Level	Meaning	Color	Description
5	Reminder	Green	Technically, a reminder is not an alert. It usually indicates that an administrator has performed an important action. For example, `the administrator deleted a cluster.`

Templates

Templates are used to generate dynamic content using variables during the runtime. They can be used for:

- Generating alerts (configuring the alert overview and details templates for alert items)
- Notifications (configuring the message and message aggregation templates for channels)
- Channel parameters (for example, configuring the Header and Body content templates in the HTTP channel)

Sample template:

- alarm_summary:

  ```
  The CPU utilization of ${alarm_targe} exceeds the threshold.
  ```

- alarm_description:

  ```
  The CPU utilization of ${alarm_targe} exceeds the threshold of ${alarm_threshold} for ${alarm_duration}
  ```

  .

For more information about the supported template variables, see Appendix 4. OCP alert template variables.

Alert aggregation

To avoid alert storms caused by too many alerts, you can aggregate the alert channels.

Aggregation rules:

- Aggregate OceanBase log alerts by alert type, log error code, and OceanBase cluster.
- Aggregate other OceanBase alerts by alert type and OceanBase cluster.
- Aggregate application alerts by alert type and alert target.

Alert clearance

An alert is cleared when a fault is recovered. After that, the monitoring module identifies that the fault is resolved and notifies the alert service, or the alert service clears the alert automatically after the clearance timeout expires.

The logic of clearing alerts after a timeout period:

- Each alert item has a check cycle and ignorance cycle.
- During the new check cycle, the monitoring module calls the alert API to set the alert item as cleared if it finds that the alert item meets the clearance criteria.
- When the ignorance cycle expires, the alert item is considered cleared if the alert item is no longer reported.

17.1.3.4.3. Configure alert items

17.1.3.4.3.1. Create an alert item

This topic describes how to create an alert item.

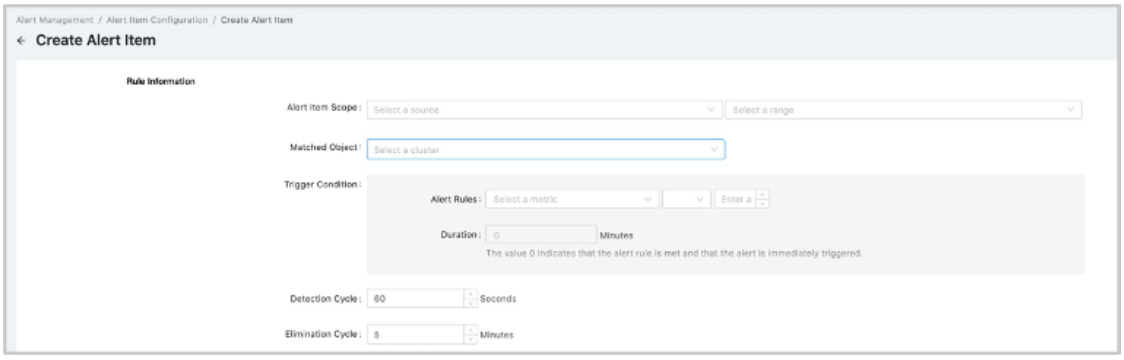
Prerequisites

The current logon user are granted the ADMIN or ALARM_MANAGER role.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management** > **Alerts**.
3. On the Alert Item Configuration tab, click **Create Alert Item** in the upper-right corner.
4. Configure the Rule Information for the alert item.
   i. Configure the Alert Item Scope.
   ii. Configure the Matched Object to determine the objects to which the alert rule applies.
   iii. Configure the Trigger Condition.
   - You can select different metrics from the Alert Rules drop-down list based on the scope of alert item.
   - 0 in the Duration field indicates that the alert is immediately triggered. You can set the duration to prevent false positives caused by indicator glitches.

   iv. Configure the detection cycle and elimination cycle.



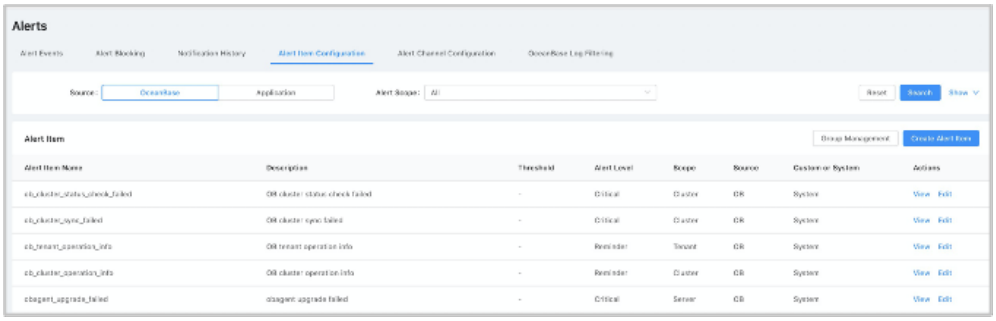
5. Configure the basic information of the alert item.
   i. Configure the Alert Item Name and Chinese name.
   ii. Configure the Alert Level.
   iii. Specify the Alert Overview Template as prompted.
   iv. Specify the Alert Details Template as prompted.
   - The alert overview template is referenced in the template field of a channel and corresponds to alarm_summary.

- The alert details template is referenced in the template field of a channel and corresponds to alarm_description.

6. Click **OK**.

17.1.3.4.3.2. View alert items

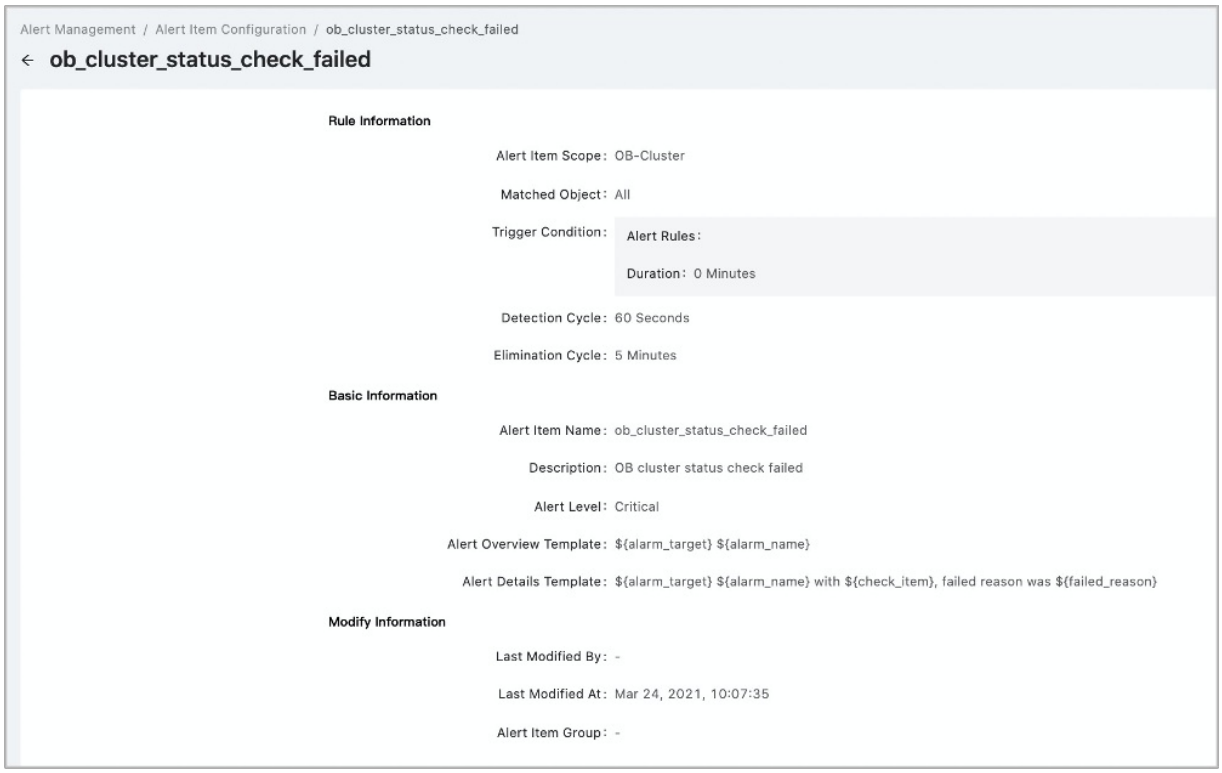
In the left-side navigation pane, choose System Management > Alerts. On the Alert Item Configuration tab, you can view all the existing alert items.



An alert item is displayed in each row of the alert item list. You can view, edit, or delete an alert item in the rightmost **Actions** column.

> **Note**
>
> System alert items cannot be deleted. The edit operation only allows you to modify only a limited number of configuration items, such as the threshold, duration, and alert level.

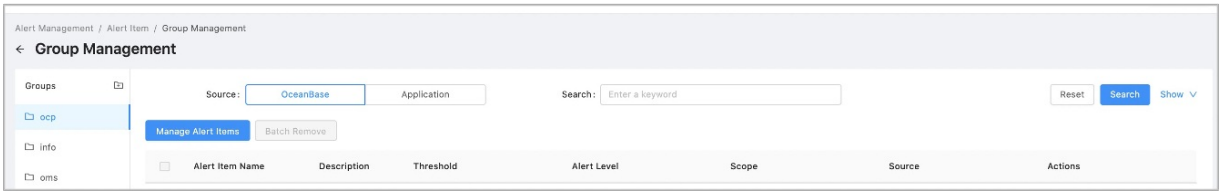
Click **View** to view the detailed configuration information of an alert item.



17.1.3.4.3.3. Group alert items

OCP allows you to classify alert items and assign them into different groups. In this case, you can subscribe to alert notifications in a simple way.

In the left-side navigation pane, choose **System Management > Alerts**. On the **Alert Item Configuration** tab, click **Group Management** to view the groups of all alert items. You can click a group name to view the alert items in the group.



Manage a group

- Retrieve alert items: OCP allows you to use multiple methods to retrieve alert items. For example, you can search for alert items by **source, keyword, application, alert level**, and **alert scope**.
- Group alert items: In the Groups section, click the group name to view the alert items in the group. On the page that appears, click **Manage Alert Items**, and add the alert items in the left-side section to the right-side section.
- Remove alert items from a group: In the Groups section, click the group name to view the alert items in the group. On the page that appears, select the alert items that you want to remove from the group, and click **Batch Remove**.

> **Note**
>
> This operation only removes the alert items from the group, and the alert items are not deleted.

- Create a custom group: In the Groups section, click the Create icon next to Groups, and enter the name of the new group.
- Change a group name: Find the group name that you want to change, click the More icon on the right, and click **Rename**.
- Delete a group: Find the group name that you want to delete, click the More icon on the right, and click **Delete**.

Default groups

The system provides four default groups. You can edit these groups, but cannot delete them.

Group name	Description
dba	The alert items that the database administrator must pay attention to.
dev	The alert items that the developers must pay attention to.
oms	The alert items that the administrator of OceanBase Migration Service (OMS) must pay attention to.
backup	The alert items that the administrator of backup and recovery must pay attention to.
info	The reminders, such as a reminder of O&M operation results.
ocp	The alert items for OCP basic modules, such as host alerts.

17.1.3.4.3.4. Edit an alert item

System administrators and OCP alert administrators can modify an alert item based on business requirements.

Prerequisites

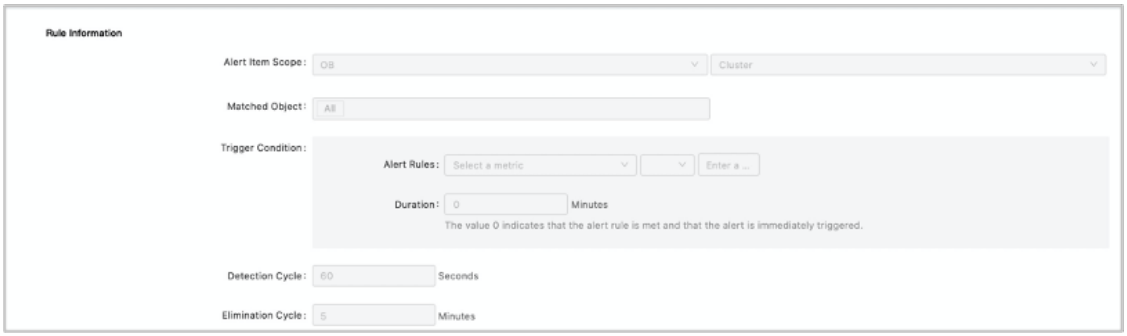
The current logon user is granted the system administrator and OCP alert management roles.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Alerts**.
3. On the **Alert Item Configuration** tab, find the alert item that you want to edit.
4. Reset the Rule Information of the alert item.
   i. Configure the **Alert Item Scope**.
   ii. Configure the **Matching Objects** to determine the objects to which the alert item applies.
   iii. Specify the **Trigger Condition**.
   - You can select different metrics from the Alert Rules drop-down list based on the scope of the alert item.

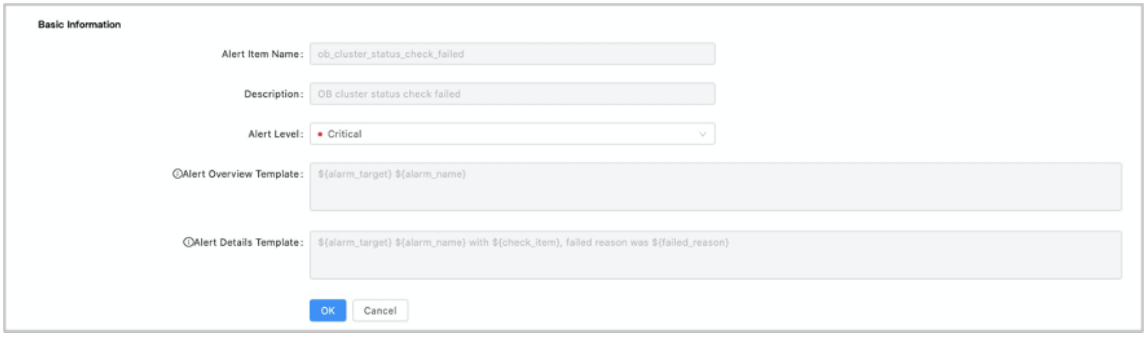
- The value **0** in the Duration field indicates that the alert is immediately triggered. You can specify the duration to avoid false positives due to indicator glitches.

iv. Configure **Detection Cycle** and **Elimination Cycle**.



5. Configure the **Basic Information** of the alert item.
   i. Configure the **Alert Item Name** and Chinese **name**.
   ii. Configure **Alert Level**.
   iii. Specify the **Alert Overview Template** as prompted
   iv. Specify the **Alert Details Template** as prompted.

- The alert overview template is referenced in the template field of a channel and corresponds to alarm_summary.
- The alert details template is referenced in the template field of a channel and corresponds to alarm_description.



6. Click **OK**.

17.1.3.4.3.5. Delete an alert item

You can delete an alert item as a system administrator or an OCP alert administrator based on your needs. Alert items that are predefined by the system cannot be deleted.

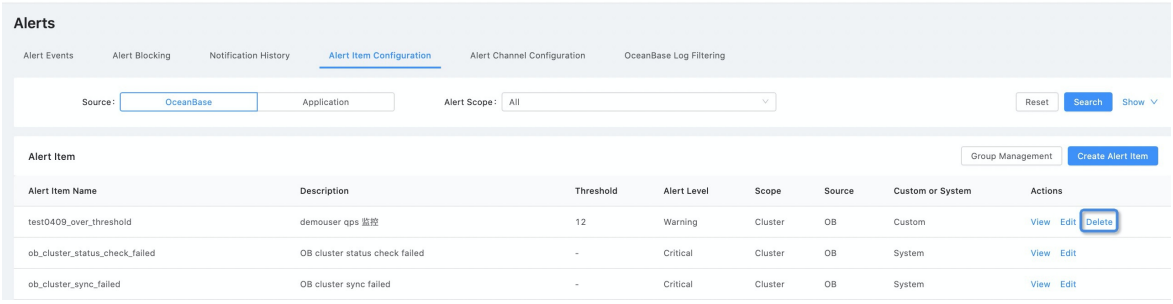
Prerequisites

- The current user you log on to is the ADMIN or ALARM_MANAGER role.
- The alert item that you delete is a custom alert item instead of an alert item that is predefined by the system.

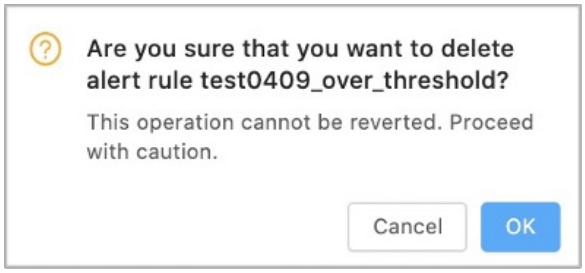
Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Alerts**.
3. On the **Alert Item Configuration** tab, find the alert item that you want to delete.

4. In the Actions column, click **Delete**.



5. In the message that appears, click **OK**.



17.1.3.4.4. Alert channel configuration

17.1.3.4.4.1. Notification channels

Alerting is an independent feature. Before you configure notification channels and subscribe to alerts, you can view alerts only on the Alert Events page of OCP. To receive alert notifications, configure notification channels and subscribe to alerts.

Notification channels are the channels through which alert notifications are sent. You can configure the following types of channels:

- HTTP
- Custom scripts

> **Note**
>
> You can configure the time parameters of the alert aggregation in the channel. After you update the notification channel, its aggregation is restarted. In this case, if a valid alert item exists, you will receive an alert notification in a short while.

17.1.3.4.4.2. Create an alert channel

This topic describes how to create an alert channel.

Alerting is a standalone feature. If you do not configure alert channels or subscriptions, you can view alerts only on the **Alert Events** page of the OceanBase Cloud Platform (OCP) console. To receive alert notifications, you must configure an alert channel and subscribe to an alert.

The alert channel allows OCP to send notifications and supports the following channel types:

- HTTP
- Custom script

> **Note**
>
> You can configure the time variables for alert aggregation when you configure a channel. After an alert channel is updated, the channel restarts alert aggregation. If a valid alert item exists, you may receive alert notifications faster.

Prerequisite

You have configured the basic information of the recipients.

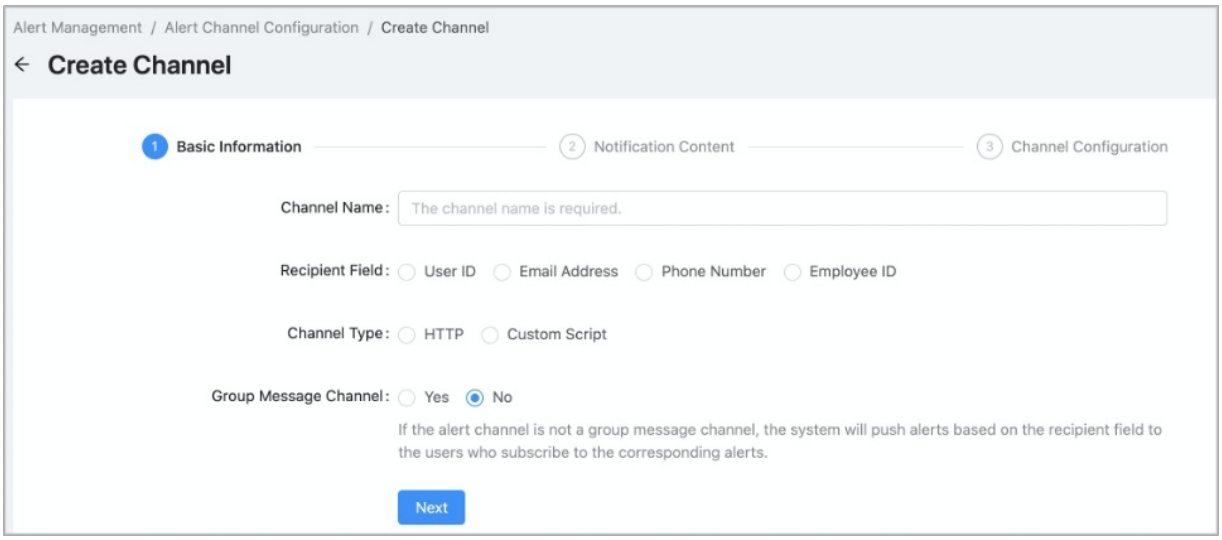
Procedure

1. Log on to the OCP console.
2. In the left-side navigation pane, choose **System Management > Alerts**.
3. On the **Alert Channel Configuration** page, click **Create Channel**.
4. i. Set the Basic Information of the channel.
   a. Enter the **channel name**.
   b. Select the **recipient field**.
   - The recipient field can be set as **User ID, Email Address, Phone Number** or **Employee ID**.
   - The recipient field is associated with the basic information of the OCP user.

   a. Set the notification type.
   - **HTTP**
   - **Custom script**

   a. Specify whether this channel is a **group message channel**.
   - If you select Yes, the system pushes notifications even when no one subscribes to the alerts.
   - If you select No, the system pushes messages to users who have subscribed to the corresponding alerts based on the Recipient Field.

   a. Click **Next**.

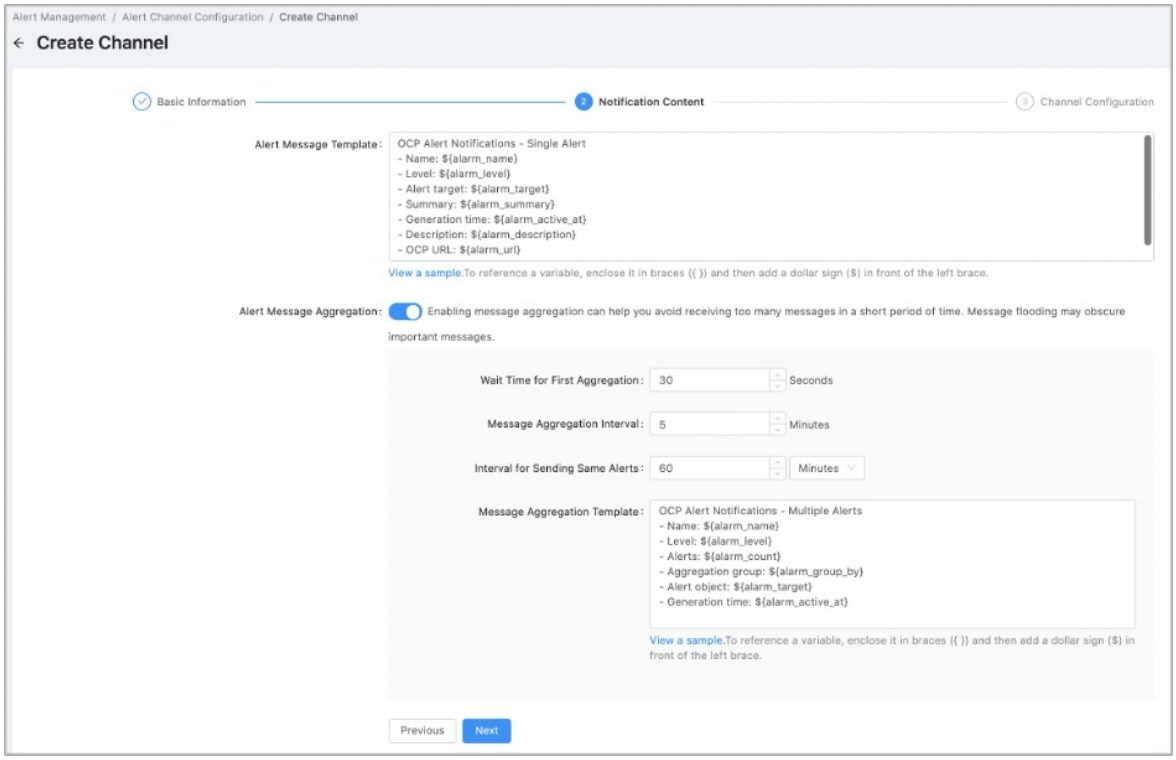


5. Edit the **notification content**.

i. Edit the **alert message template**.
   - Variables are supported in the alert message template.
   - You can customize the alert message template as prompted.

ii. Specify whether to **Enable Alert Message Aggregation**.
   - Alert message aggregation can help you avoid missing important messages when you receive too many messages in a short time. You can check the aggregation rule as prompted
   - If you enable message aggregation, you must also configure the parameters such as the aggregation message template.

iii. Click **Next**.



6. Configure the channel. Configure the parameters for the channel type that you selected. Take the HTTP channel as an example.

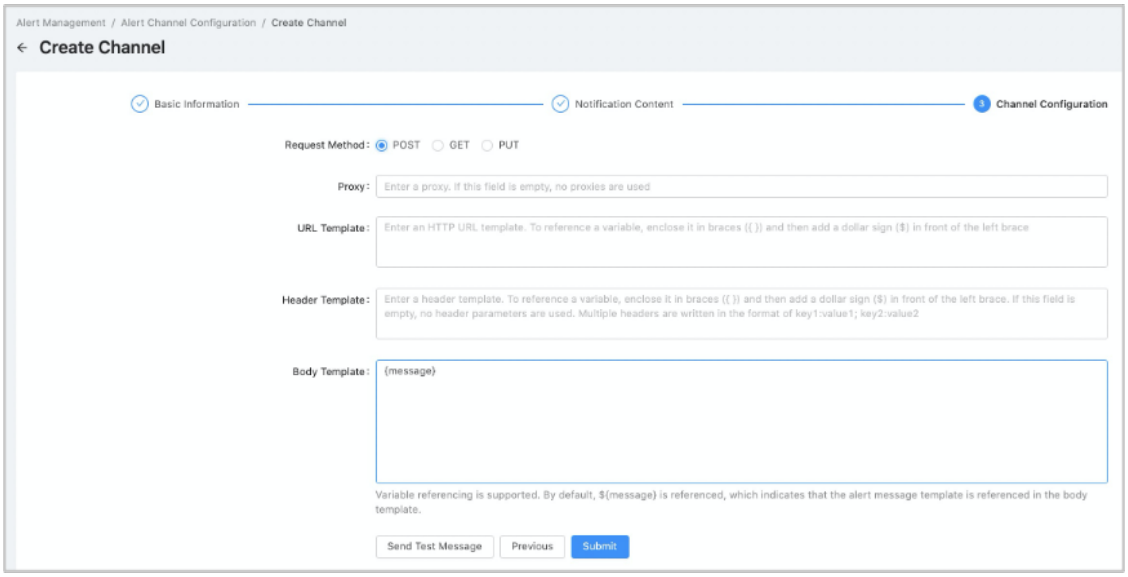
   i. Select the **request method**. The following request methods are supported: POST, GET, and PUT.

   ii. Specify the **proxy**. If you do not specify this field, no proxy is used.

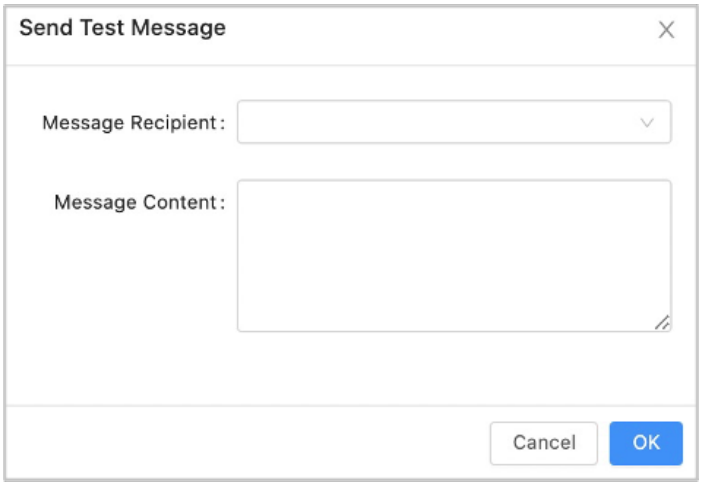
   iii. Edit the **URL template**. You can enter $ and {} to reference variables.

   iv. Edit the **header template**. You can enter $ and {} to reference variables. If you do not specify this field, no header parameter is used.

v. You can reference variables when you edit the **body template**. By default, ${message} is referenced, indicating that the alert message template is referenced in the body template.



vi. Click **Send Test Message**. If the message is sent, click **Submit**.

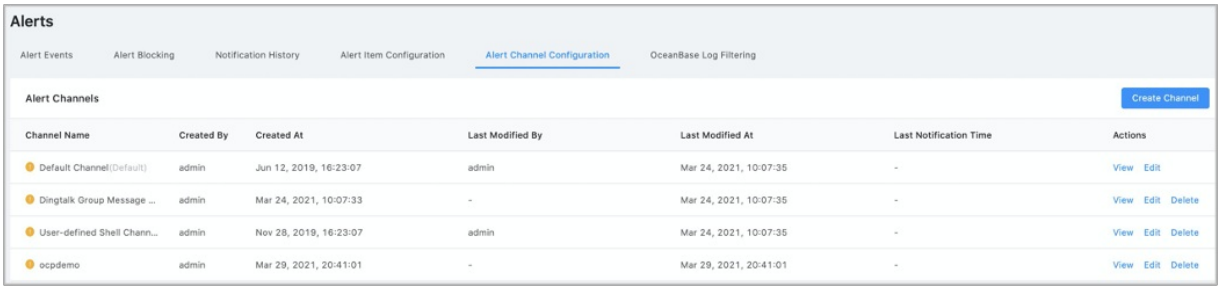


17.1.3.4.4.3. View alert notification channels

This topic describes how to view alert notification channels.

Alert notification channels

In the Alert Channels section, you can view all channels. You can edit or delete a channel as an administrator.

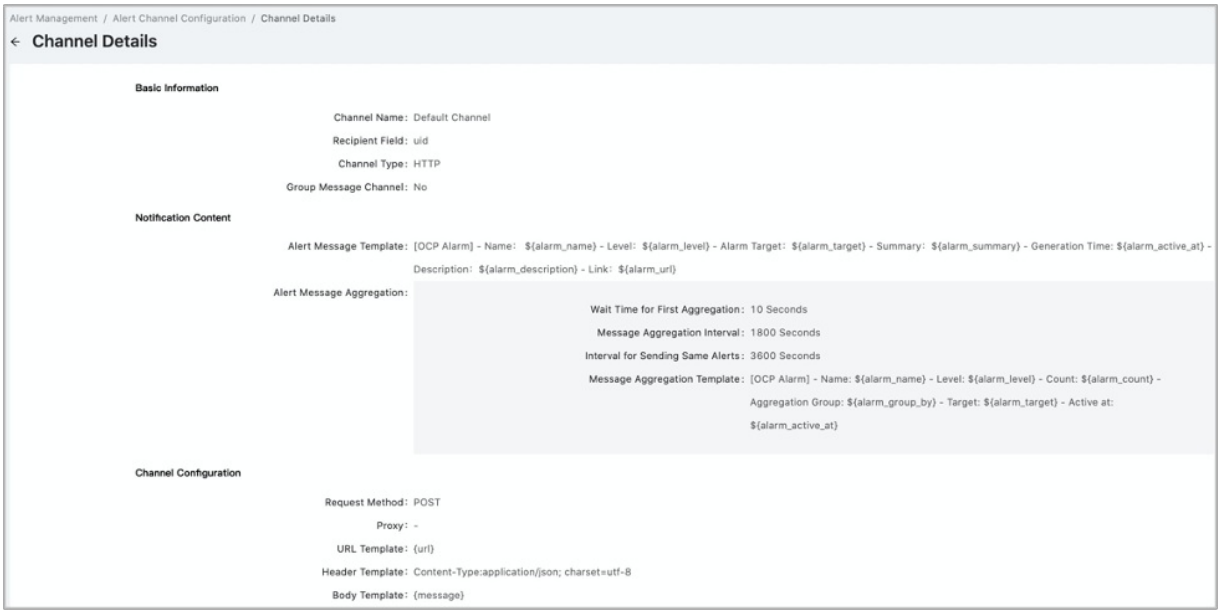


Default channel

OCP presets a default channel for O&M alert notifications. You can subscribe to the notifications. Take note of the following considerations:

- You cannot delete the default channel.
- Before you use the default channel, make sure the channel is configured.

View the configuration of the default channel:



17.1.3.4.4.4. Edit an alert notification channel

After an alert notification channel is created, you can modify the basic information, content, and configuration of the notification channel as an administrator based on business needs.

Prerequisites

An alert notification channel is created.

Procedure

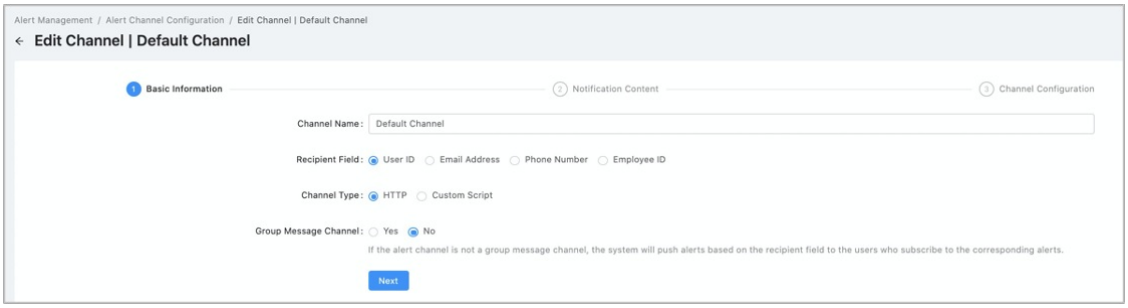
1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Alerts**.
3. On the **Alert Channel Configuration** page, find the channel that you want to modify.
4. Click **Edit** in the **Actions** column.
5. Reset **Basic Information** for the channel.
   i. Specify **Channel Name**.
   ii. Specify **Recipient Field**.
   - You can set Recipient Field to **User ID, Email Address, Phone Number**, or **Employee ID**.
   - Recipient Field is associated with the basic information of the OCP user.

   iii. Specify Channel Type.
   - **HTTP**
   - **Custom Script**

   iv. Specify **Group Message Channel**.

- If you set Group Message Channel to Yes, the system sends notifications to all group users regardless of whether a user subscribes to the notifications.
- If you set Group Message Channel to No, the system sends notifications based on Recipient Field. This field specifies the users who have subscribed to the notifications.

v. Click **Next**.



5. Reset **Notification Content**.

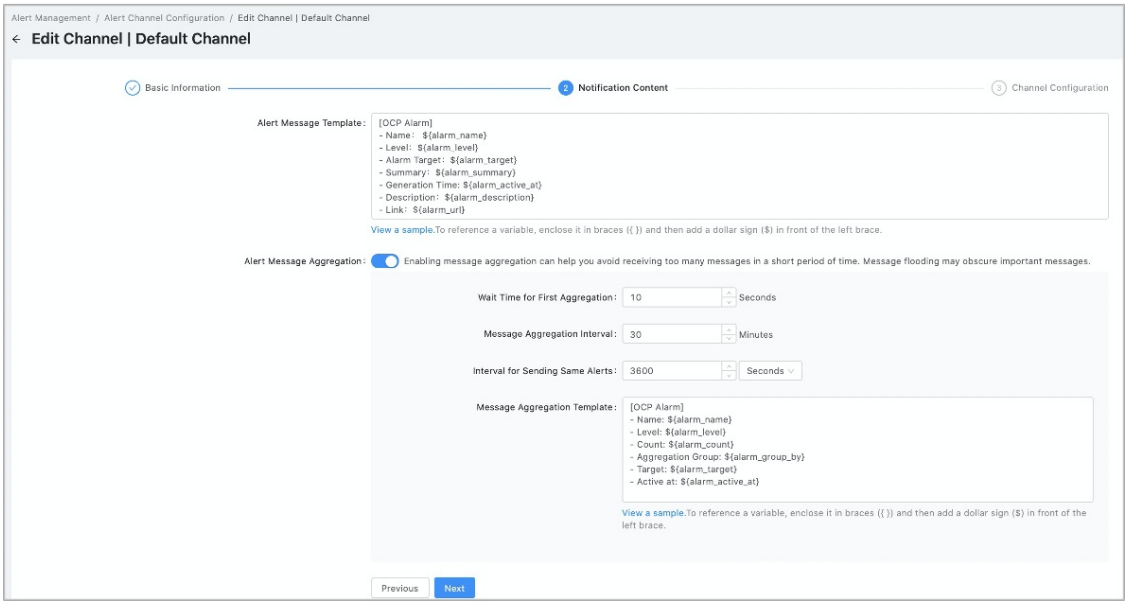
i. Specify **Alert Message Template**.

- Alert notification templates that have variables are supported.
- You can configure custom alert messages based on the instructions on the page.

ii. Specify whether to enable **Alert Message Aggregation**.

- Alert message aggregation helps you avoid receiving a large number of messages in a short period of time. Message flooding may obscure important messages. You can view the aggregation rules based on the instructions on the page.
- To enable message aggregation, configure the parameters, such as Message Aggregation Template.

iii. Click **Next**.



6. Reset **Channel Configuration**.

Configure the parameters based on the selected channel type.

If you set **Channel Type** to **HTTP**, configure the parameters based on the following description.

i. Specify **Request Method**. Valid values: POST, GET, and PUT.

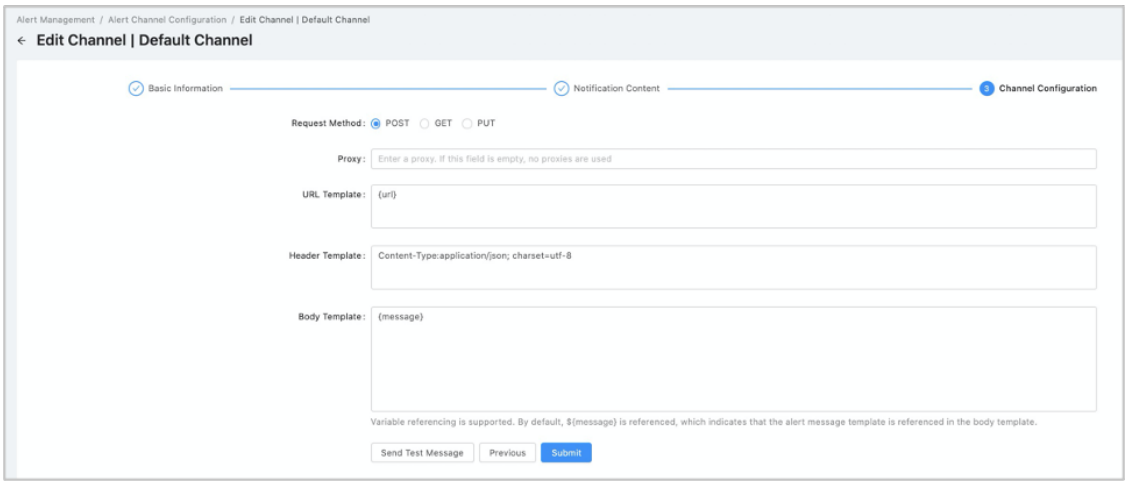
ii. Specify **Proxy**. Leave this parameter empty if no proxy is used.

iii. Specify **URL Template**. To reference a variable, enclose it by using braces ({}) and a dollar sign ($).

iv. Specify **Header Template**. To reference a variable, enclose it by using braces ({}) and a dollar sign ($). If this field is empty, no header parameters are used.

v. Specify **Body Template**. You can reference variables. By default, the **${message}** variable is referenced. This indicates that the alert message template is referenced in the body template.

vi. Click **Send Test Message**. In the dialog box that appears, select the recipient, enter the message content, and then click **OK**. If the message is sent, click **Submit**.



If you set **Channel Type** to **Custom Script**, configure the parameters based on the following description.

i. Select a configuration method for the script-based channel. If you select **Configuration Script Content**, enter the script content again. The script content must meet the following requirements:

- Only Bash or Python scripts are supported.
- The first line of the script must contain shebang that specifies the program to run the script.

For more information about sample custom scripts, see **Configure alert notification channels** > **Configuration examples** in this document.

If you set **Configuration Method** to **Configure Script Name**, rename the script file based on the instructions.

- The script file name can contain only letters, digits, underscores (_), and periods (.).
- Attackers may exploit the Script Name field to introduce shell injection. For security risks, the script name cannot contain spaces or tab characters.

b. Click **Send Test Message**. In the dialog box that appears, select the recipient, enter the message content, and then click **OK**.

c. If the message is sent, click **Submit**.

When a message is sent through a script-based channel, OCP calls the script you configure and passes all the variables related to the notification to the script through environment variables. For information about variables in message notifications, see Appendix 4 OCP alert template variables.

17.1.3.4.4.5. Delete an alert channel

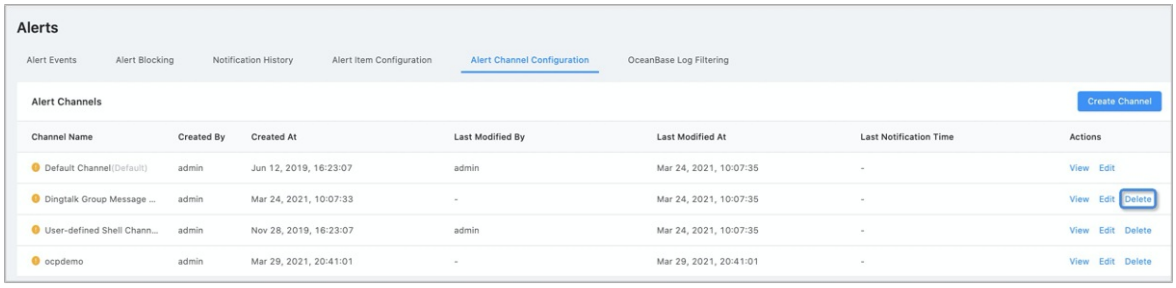
The administrator can delete custom alert channels based on business requirements.

Prerequisites

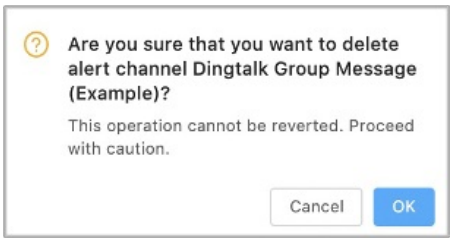
- The channel that you want to delete is not the default channel.
- The current user you log on to is the system administrator or OCP alert administrator.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Alerts**.
3. On the **Alert Channel Configuration** tab, find the channel that you want to delete.
4. In the **Actions** column, click **Delete**.



5. In the message that appears, click **OK**.



17.1.3.4.4.6. Configuration examples

This topic describes some examples on how to configure an alert channel.

DingTalk group messages

Group messages can be sent through the webhook URL of a DingTalk chatbot. For more information about DingTalk chatbot, see Official documentation of DingTalk chatbot.

Procedure

1. Configure the DingTalk chatbot. To configure a chatbot for a DingTalk group, you must record the Webhook URL. To use DingTalk chatbot, you must configure the security settings. OCP supports only the following two types of security settings for the HTTP alert channels: **Custom Keywords** and **IP Addresses**. The following figure shows the page of DingTalk chatbot security settings.

   - Set the **Custom Keywords**. Enter one or more keywords in the custom keywords field that match the message template configured for the OCP alert channel. If the first line of the default template contains "OCP alert notifications", you can set the custom keywords as "OCP alerts".
   - Configure the **IP Addresses**. Note that the IP address you configure must be the public IP address of the OCP server. You can run the following command to view the public IP address of the server: curl -w "\n\n" -s http://whatismijnip.nl. If the server does not have a fixed public IP address, contact your network administrator to obtain the public IP CIDR block of the server.
2. Create an alert channel in the alert channel configuration page. Set the channel type as **HTTP** and select **Mobile Number** for the recipient field. Select **Yes** for the group message channel field.
3. Configure the notification content.
   - Message template

```
OCP Alert Notification - Single Alert
- Name: ${alarm_name}
- Level: ${alarm_level}
- Alert target: ${alarm_target}
- Summary: ${alarm_summary}
- Generation time: ${alarm_active_at}
- Description: ${alarm_description}
- OCP URL: ${alarm_url}
```

- Message aggregation template

```
OCP Alert Notification - Multiple Alerts
- Name: ${alarm_name}
- Level: ${alarm_level}
- Alerts: ${alarm_count}
- Aggregation group: ${alarm_group_by}
- Alert target: ${alarm_target}
- Generation time: ${alarm_active_at}
```

4. Alert channel configuration
   - Request Method: POST.
   - Header Template: `Content-Type:application/json; charset=utf-8`

     > **Note**
     > The URL template is the Webhook URL recorded in step 1.

   - Header Template: `Content-Type: application/json; charset=utf-8`
   - Body template:

```
{
  "msgtype": "text",
  "text": {
    "content": ${message_json}
  },
  "at": {
    "atMobiles": ${recipients_json_array},
    "isAtAll": false
  }
}
```

> **Note**
> The format of the Body Template is JSON. To reference the variables at the end of the `_json` and the `json_array`, do not enclose them in `"`. This is because the values of these variables are processed by JSON's serialization. Enclosing these variables in `"` results in JSON format error. For comparison, when you reference a variable not in JSON format, you must enclose it in `"`. For example, `"{message}"`.

DingTalk group messages in Markdown format

DingTalk messages in Markdown format is available. In the following examples, messages in rich text format (RTF) are configured by using the formatting capability of Markdown. The configuration method is the same as that for text format messages. However, several templates are configured in different ways. For example, the font color of the message in RTF is configured by referencing

```
${alarm_level_color}
```

For more information about the font colors of alert messages in different levels.

For more information about the message types that DingTalk supports, see DingTalk development documentation.

- Message template

```
<font color=${alarm_level_color} > OCP Alert Notification - Single Alert</font>
- Name: ${alarm_name}
- Level: ${alarm_level}
- Alert target: ${alarm_target}
- Summary: ${alarm_summary}
- Generation time: ${alarm_active_at}
- Description: ${alarm_description}
- [View on OCP](${alarm_url})
```

- Message aggregation template

```
<font color=${alarm_level_color} > OCP Alert Notification - Multiple Alerts
- Name: ${alarm_name}
- Level: ${alarm_level}
- Alerts: ${alarm_count}
- Aggregation group: ${alarm_group_by}
- Alert target: ${alarm_target}
- Generation time: ${alarm_active_at}
```

- Body template

```
{
    "msgtype": "markdown",
    "markdown": {
        "title": "test of DingTalk alert messages in Markdown format in OCP",
        "text": ${message_json}
    }
}
```

Custom script channel

Custom script channels are designed for special alert recipients. In most cases, the custom script channels are used in scenarios where alerts cannot be configured in HTTP format or alert have special requirements in format. You can customize some special logic in a script. For example, you can convert the alert time and add the default parameters of the third-party alert gateway through custom script channels.

1. Configure the basic information. Select **Custom Script** for channel type. Select options in other fields based on your needs.
2. Configure the notification content.
   - Message template

```
[OCP alert notification]
- Name: ${alarm_name}
- Level: ${alarm_level}
- Alert target: ${alarm_target}
- Summary: ${alarm_summary}
- Generation time: ${alarm_active_at}
- Description: ${alarm_description}
- OCP URL: ${alarm_url}
```

- Message aggregation template

```
[OCP alert notification]
- Name: ${alarm_name}
- Level: ${alarm_level}
- Alerts: ${alarm_count}
- Aggregation group: ${alarm_group_by}
- Alert target: ${alarm_target}
- Generation time: ${alarm_active_at}
```

3. Configure alert channels.

- Script Name: Select when you configure the channel **Set the Configuration Method** as **Configure Script Name**.

By default, the system provides a sample custom script: alarm_send_script_demo.sh. You can modify the sample custom script or add a custom script. When you modify or add a script file, make sure that the script file is stored in the /home/admin directory of each OCP.

Currently, only Bash or Python scripts are supported.

- **Shell script sample**

Send DingTalk group messages:

```
#! /bin/sh

echo "Below is a list of alarm paras"

# below variables can be referenced by prefix "$", for example, $alarm_name or ${alarm_name}
echo "alarm_name:$alarm_name"
echo "app_type:$app_type"
echo "alarm_threshold:$alarm_threshold"
echo "alarm_time=$alarm_time"
echo "alarm_last_interval:$alarm_last_interval"
echo "alarm_time:$alarm_time"
echo "alarm_level:$alarm_level"
echo "alarm_type:$alarm_type"
echo "alarm_summary:$alarm_summary"
echo "alarm_url:$alarm_url"
echo "app:$app"
echo "alarm_duration:$alarm_duration"
echo "alarm_status:$alarm_status"
echo "alarm_scope:$alarm_scope"
echo "alarm_active_at:$alarm_active_at"
echo "alarm_target:$alarm_target"
echo "alarm_description:$alarm_description"
echo "message:$message"
echo "receiver:$receiver"
echo "alarm_id:$alarm_id"

# this function defines to how to assembly request by yourself according to your requirements
# this demo shows you how to send alarm to ding ding
function send(){
    # this token is ding ding group token, please apply and assign it to variable token
    token=''
    # URL="https://oapi.dingtalk.com/robot/send?access_token=$token"
    URL=''
    curl -X POST ${URL} -H 'Content-Type: application/json' -d '{"msgtype":"text","text":{"content":"
'"${message}"'"}}'
}

# invoke function to
send
```

- **Python script sample**

Output alert information to a text file:

```
#! python
# -*- coding: utf-8 -*-
import os

# Obtain the values of the environment variables
message = os.environ['message']
print "hello {}".format(message)
with open("./sample.txt", "a") as file_object:
    file_object.write("{}".format(message))
```

When you write a script, note the following items:

- The return code exitCode of a script is used to determine whether the execution is successful.

`exitCode==0`

Indicates that the script execution succeeds. A non-zero value indicates that the execution fails.

- If a script fails to be executed, the error information can be sent to stderr. OCP records the error message for subsequent troubleshooting.
- Scripts should be executed as soon as possible. We recommend that execution time be controlled within 1 second. OCP sets the time-out period for a single script calling. The default time-out period is 10 seconds. If the output is not returned within 10 seconds, OCP forces the script process to stop.
- When you write scripts, do not use multi-process programming. This can prevent the message accumulation when OCP fails to stop a script process.

- Script Content: Select when you configure the channel **Set the Configuration Method** as **Configure Script Content**.

You can customize the channel of the script. You can specify both the name and the content of a script. You can directly enter a Bash or Python script in the **Script Content** field.

4. Send test messages.

When you debug a script, you can send test messages. By default, the test messages include the variable ${message} in the script. Make sure that the variable is in the body of the test message that you send. The logic to determine whether a custom script is successfully sent depends on the specific situation. If the system considers that the script execution succeeds, the script is successfully sent. However, in the developer mode of the browser, users can debug a script based on the response content of the _test request corresponding to the test message that you sent.

17.1.3.4.5. Alert subscription settings

By default, alerts can be viewed only on the Alerts page. Users can subscribe to alerts only after the administrator has configured an alert channel.

You can receive alert notifications only when the following two conditions are met:

1. The administrator has configured an alert channel.
2. You have subscribed to an alert.

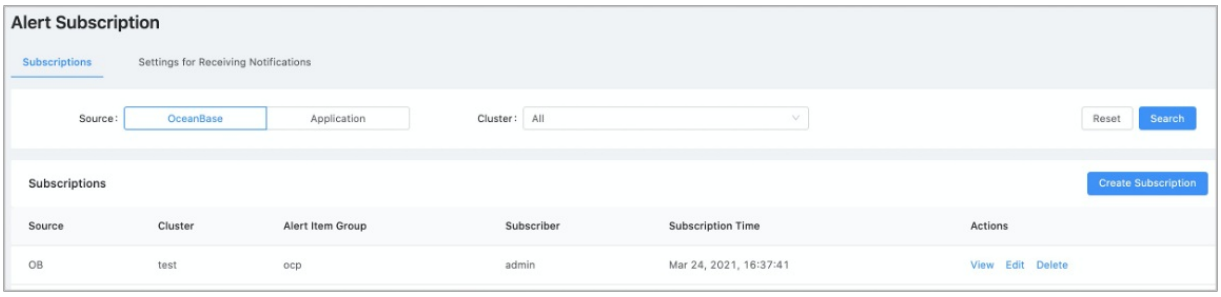
To send alert notifications from OCP, perform the following steps:

1. Configure an alert channel. The administrator configures the alert channel.
2. Create a new subscription. You can configure the subscriptions.
3. Configure the settings for receiving notifications. You can configure whether to receive notifications and the reception limits.

View alert subscriptions

Log on to the OCP console. Expand the hidden list in the User Center in the upper-right corner, and click **Alert Subscription**. On the Subscriptions tab, you can filter alerts by **Source** and **Cluster**.

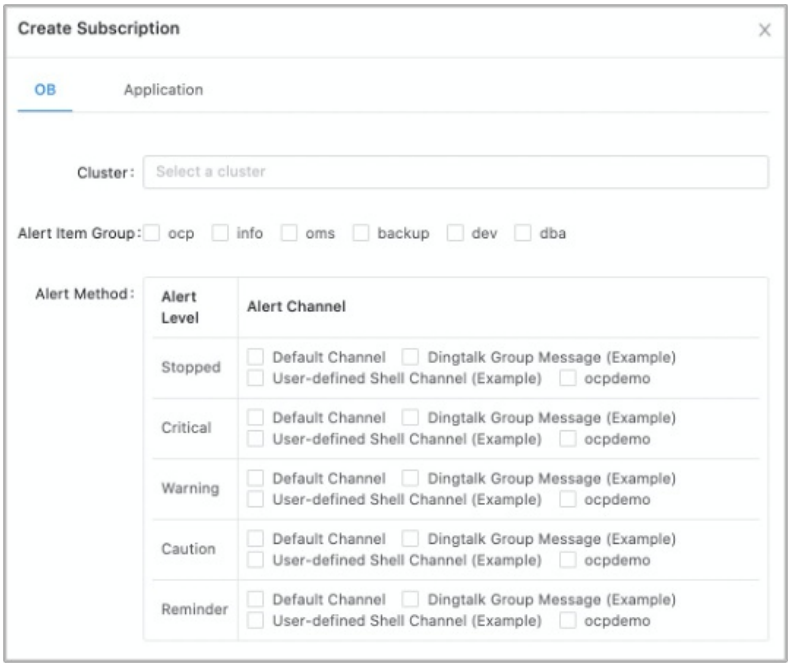
By default, the page displays the **Source, Cluster, Alert Item Group, Subscriber** and **Subscription Time**.



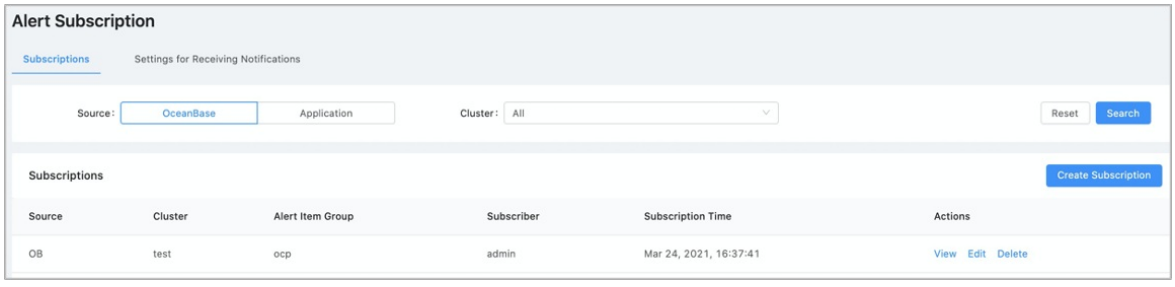
Create an alert

> **Note**
>
> You can only select a cluster that you have the read permission when you configure an alert subscription. If your read permission for a cluster is revoked after you configure an alert subscription, you cannot receive alert messages from this cluster even though the alert subscription configuration still exists.

1. Log on to the OCP console.
2. Expand the hidden list in the User Center in the upper-right corner, click **Alert Subscription**.
3. Click **Create Subscription**.
4. Click the **OceanBase** or **Application** tab and specify the OceanBase cluster or application as needed.
5. Select the **alert item groups** that you want to subscribe to.
6. Select the **alert method**. Set the alert channel for alerts of each alert level. You can configure multiple channels for each level at the same time and you must configure channels for at least one alert level.



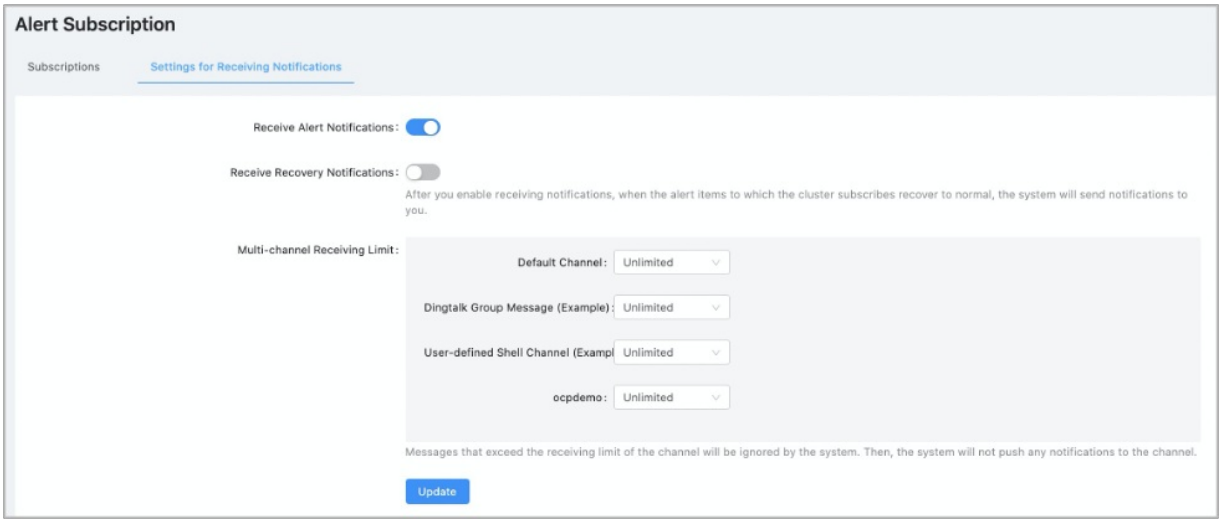
7. Click **OK**. After you complete the alert subscription settings, you can view the details and edit or delete subscriptions on the **Subscriptions** page.



Settings for receiving notifications

You can configure the settings for receiving notifications.

You will also receive fault recovery notifications for alerts that you have subscribed to. Fault recovery notifications cannot be aggregated. To avoid receiving too many such notifications, we recommend that you disable fault recovery notifications. The following table lists the setting items for receiving notifications. You can customize the settings as needed.



17.1.3.4.6. View alert events

This topic describes how to view the details of an alert event.

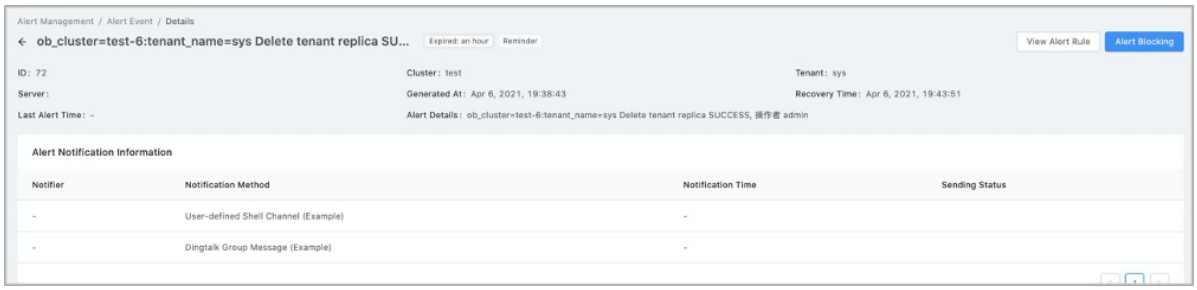
Alert event list

You can view and query all events in the alert event list. The alert event list supports multiple search conditions. You can enter a keyword to match the alert overview, alert details, and values of all tags. For example, If you set Source to OceanBase, you can enter a keyword to match the values of tags. The tags include cluster group, cluster, tenant, and host IP address.

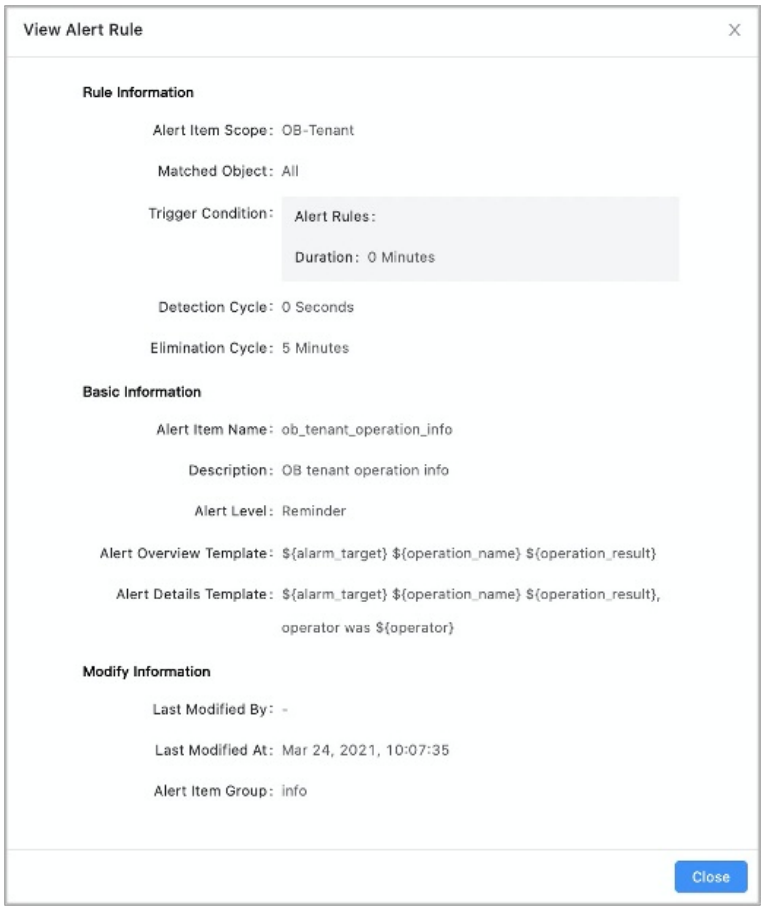
Alert event details

In the alert event list, click an alert to go to the details page of the alert event. Then, you can perform the following operations:

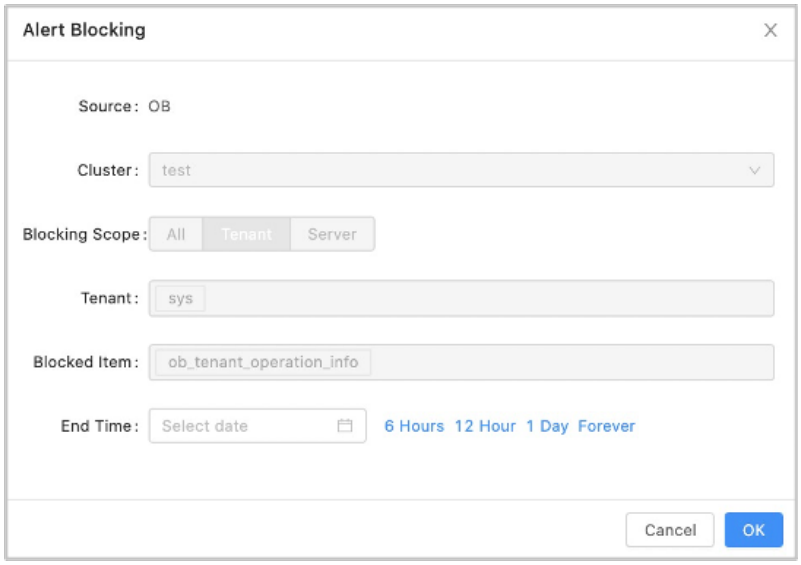
- View the details of the alert event.



- Click **View Alert Rule** in the upper-right corner to view the alert rule.



- Click **Alert Blocking** in the upper-right corner to block the alert.



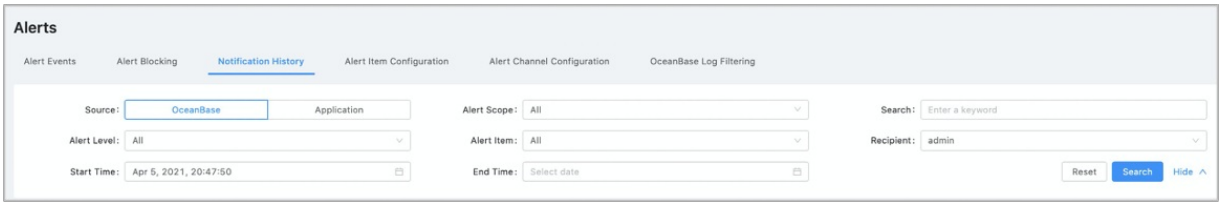
- View the notification records of the alert.

17.1.3.4.7. View the alert notification

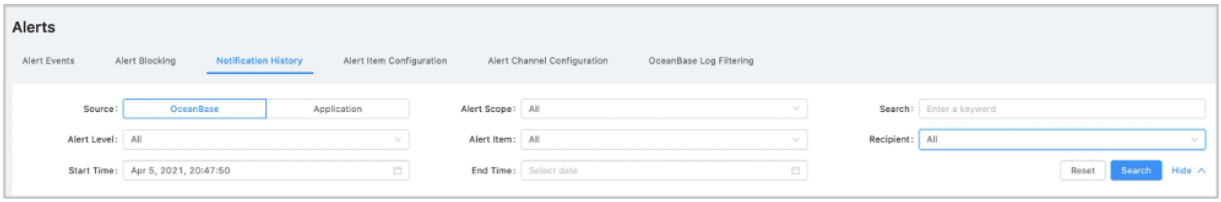
The Notification History tab displays the notification delivery records and the state of each notification. You can only view notification records in the last 90 days. Notifications older than 90 days are automatically archived.

You can view the message content, message form, notification channel, source, cluster, alert time, and recipient of a specified message.

By default, the current user is specified for the **Recipient** field as a filter condition.



> **Note**
>
> To view the alerts sent through a group message channel, select **All** from the **Recipient** drop-down list as a filter condition.

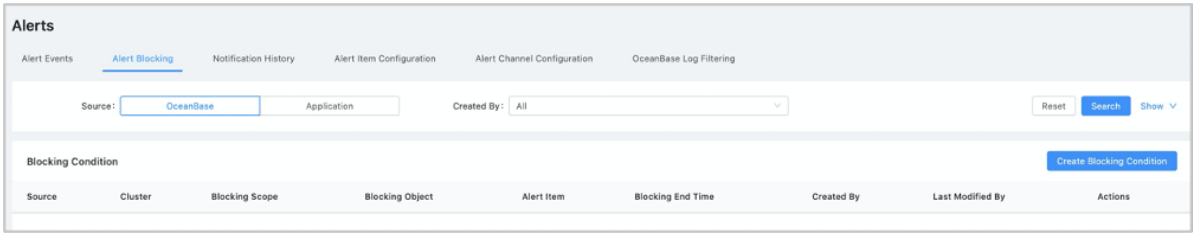


17.1.3.4.8. Block alert notifications

This topic describes how to block alert notifications.

The Alert Blocking tab displays all configurations for blocking alert notifications. By default, the configurations created by the current user appear.

- After you configure settings to block alert notifications, wait up to 30 seconds for the settings to take effect.
- The configurations for blocking alert notifications have a validity period. The system automatically archives the configurations that have exceeded the validity period more than seven days.
- When you perform an O&M task for an ApsaraDB for OceanBase cluster, the system automatically blocks alert notifications for the O&M task.



Procedure

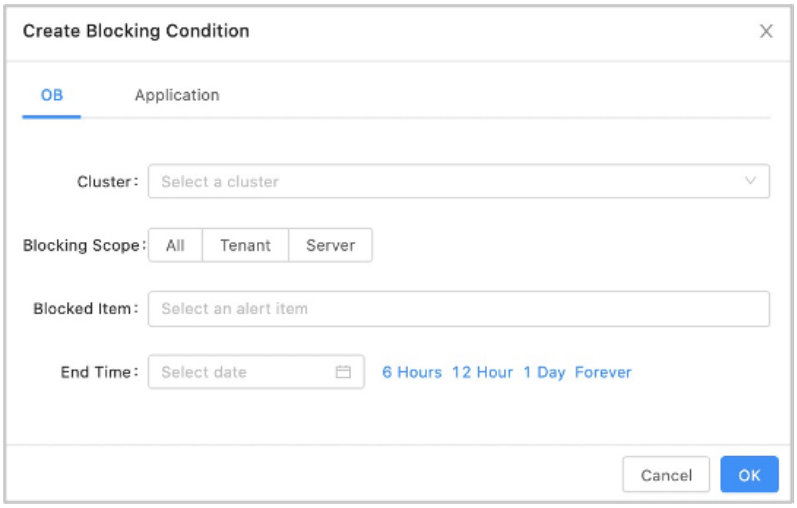
You can configure settings to block alert notifications in a few clicks on the Alert Events and Notification pages. You need to specify a validity period for the settings. You can perform the following steps to block alert notifications on the **Alert Blocking** page.

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Alerts**.
3. On the Alert Blocking tab, click **Create Blocking Condition**.
4. Click the **OB** tab or **Application** tab based on your needs. The following examples are based on the **OB** tab.
    i. Select an ApsaraDB for OceanBase cluster. You must select one cluster.

ii. Specify **Blocking Scope**.

iii. Specify **Blocked Item**.

iv. Specify **End Time**.



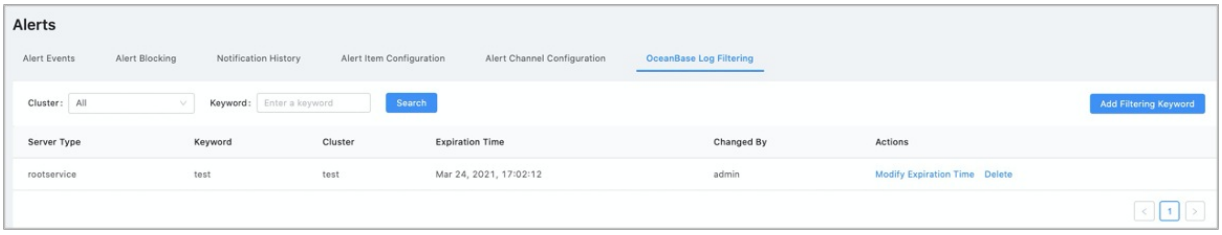
5. Click OK.

17.1.3.4.9. OceanBase log filtering

OceanBase log alert is not triggered based on the alert rule. The implementation principle is that the OBServers monitor logs. If a log of ERROR level is found, the alert is triggered. If false positives occur, you can set log filtering rules on the OceanBase Log Filtering tab.

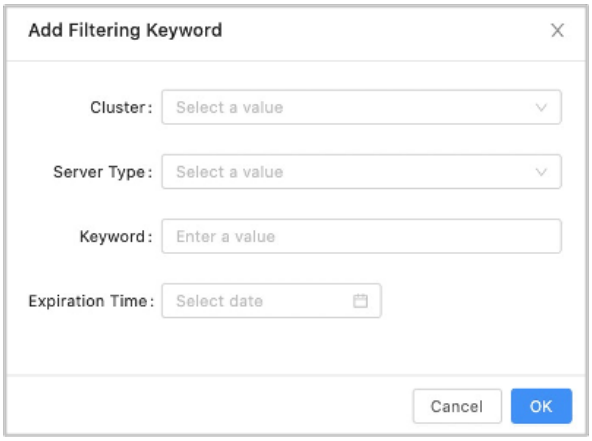
View keyword filtering rules

The **OceanBase Log Filtering** tab displays the configured log filtering rules.



Add filtering keyword

In the upper-right corner, click Add Filter Keyword, and specify the cluster, server type, keyword, and filter time. Keywords support the syntax of regular expression functions. Logs that match the keywords do not trigger alerts.



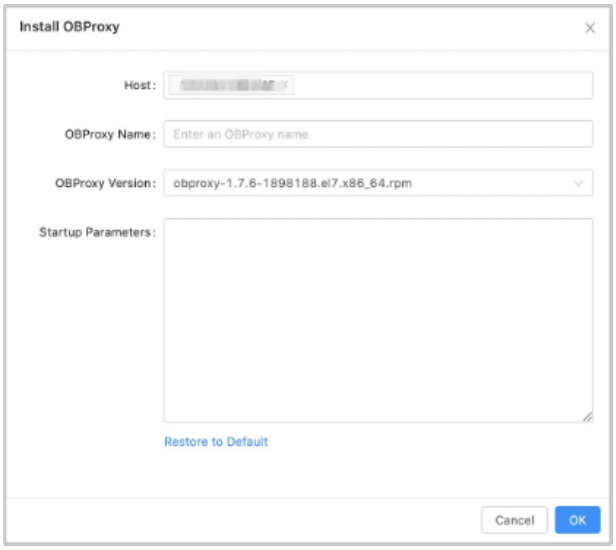
17.1.3.5. OBProxy O&M

17.1.3.5.1. Install OBProxy

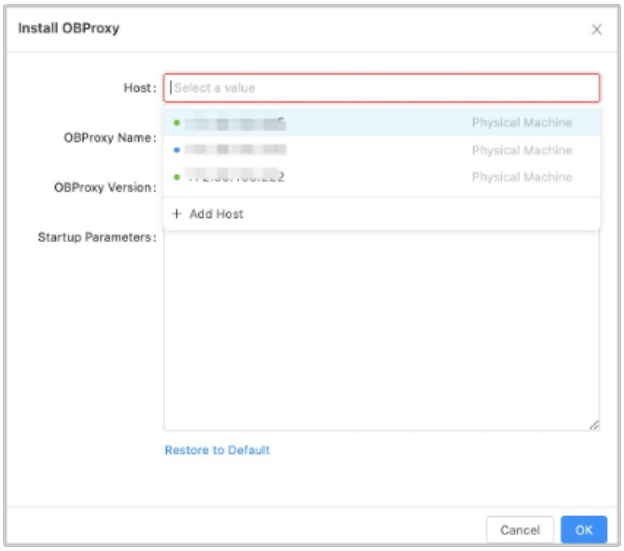
In the upper-right corner of the page, click Return to Old Version to go to the OBProxy page and install OBProxy.

Procedure

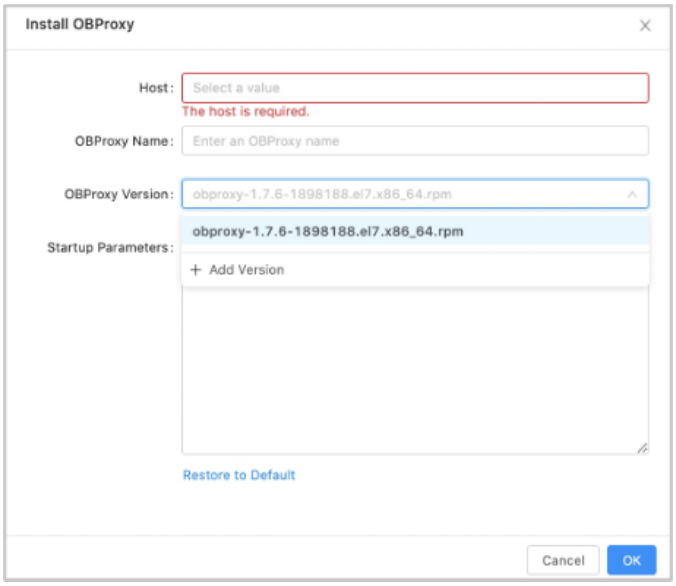
1. In the navigation pane, choose **Maintenance** to go to the **OBProxy** page.
2. Click **Install OBProxy**.
3. In the dialog box that appears, specify the following parameters:



- **Host**: Select a host from the drop-down list, or click **Add Host** to add a new host.



- OBProxy Version: Select an OBProxy version from the drop-down list, or click **Add Version** and click **Upload** in the dialog box that appears, to upload the corresponding OBProxy file.



4. Click **OK**. This will generate an OBProxy operations task. You can choose **System Management > Tasks** to check the installation progress.

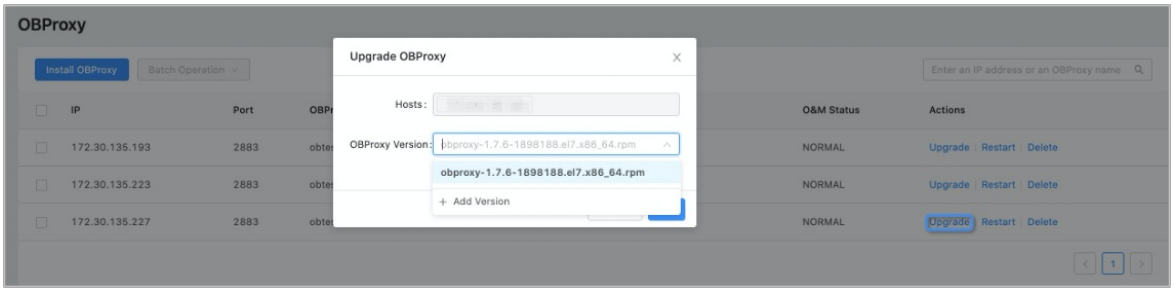
17.1.3.5.2. Upgrade OBProxy

In the upper-right corner of the page, click Return to Old Version to go to the OBProxy page and upgrade OBProxy.

Procedure

1. In the navigation pane, choose **Maintenance** to go to the OBProxy page.
2. Click **Upgrade** on the right of the IP address of the corresponding host.

3. In the dialog box that appears, select the target version, or click **Add Version** to add an OBProxy version.



4. Click **OK**. You can choose **System management> Tasks** to check the upgrade progress.

17.1.3.5.3. Restart an OBProxy

In the upper-right corner of the interface, click Back to Old Version to restart a running OBProxy.

Procedure

1. In the left-side navigation pane, click **O&M**. The OBProxy page appears.
2. Select the OBProxy that you want to restart and click **Restart** next to it.

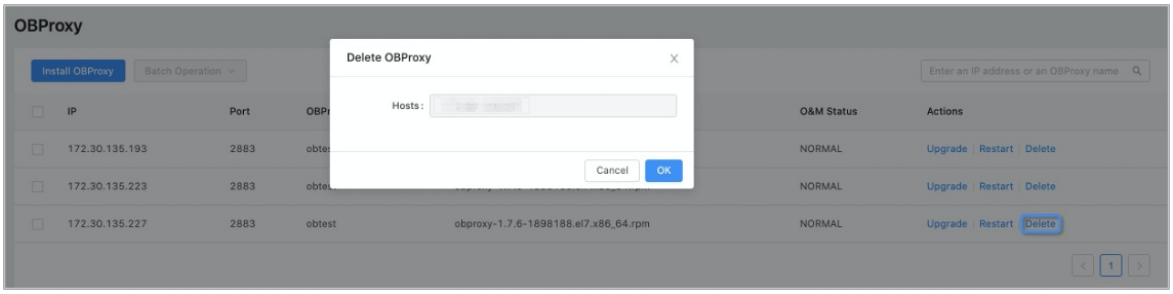
3. Click **OK**. You can view the progress by choosing **System Management > Tasks**.

17.1.3.5.4. Delete OBProxy

In the upper-right corner of the page, click Return to Old Version to go to the OBProxy page and delete redundant OBProxy.

Procedure

1. In the navigation pane, choose **Maintenance** to go to the OBProxy page.
2. Click **Delete** on the right of the corresponding instance.



3. Click **OK**. To view the deletion progress, choose **System Management > Tasks**.

17.1.3.5.5. Startup parameters

This topic describes the startup parameters.

When you install the OBProxy, you must specify startup parameters in the dialog box that appears. The startup parameters fall into the following two types:

- Startup parameters that take effect immediately after the update.
- Startup parameters that do not take effect until the OBProxy is restarted.

Parameters that take effect immediately after the update

name	info	default	range	need reboot
`mysqlclient_query_timeout`	The query timeout period for the MySQL client in the OBProxy to connect to the OBServer.	120s	[1s]	false
`mysqlclient_net_wait_timeout`	The query timeout period that the proxy sets when it uses the MySQL client to connect to the OBServer, `read_timeout`, and `write_time_out`. The value -1 indicates that the operation never times out.	10s	[-1,]	false
`mysqlclient_connection_refresh_interval`	The interval for refreshing the MySQL connection pool in the OBProxy. The actual configured time is 50 times the value	200ms	[200ms,]	false
`mysqlclient_connection_pool_warn_time`	The amount of time required for generating an alert in the connection pool. If a connection is occupied for a long period of time, an alert log is generated.	1s	[0s,]	false

name	info	default	range	need reboot
mysqlclient_conn_limit_per_observer	The maximum number of SQL connections that can be processed by each connection pool.	10	[1,]	false
proxy_info_check_interval	The interval of proxy detection of configuration updates	20s	[1s,1h]	false
schema_refresh_interval	The interval for refreshing the schema.	20s	[1s,1h]	false
server_state_refresh_interval	The interval for refreshing the OBServer status.	20s	(0s,1h]	false
config_server_refresh_interval	The interval for refreshing the config server.	20s	[60s,1d]	false
stat_sync_interval	The interval of the tasks to synchronize global statistics. The value of 0 indicates that the task is suspended	60s	[0s,1d]	false
stat_dump_interval	The interval for dumping global task data.	6000s	[0s, 1d]	false
stat_table_sync_interval	The interval of the tasks to update the global items of an internal table	60s	[0s,1d]	false
fetch_proxy_bin_random_time	The maximum amount of random time that is waited to fetch a new binary file during a hot upgrade of the OBProxy.	300s	[1s,1h]	false

name	info	default	range	need reboot
`hot_upgrade_rollback_timeout`	The default amount of time that is waited to perform a rollback after a hot upgrade of the OBProxy.	24h	[1s,30d]	false
`hot_upgrade_failure_retries`	The number of retries after a hot upgrade failure.	5	[1,20]	false
`fetch_proxy_bin_timeout`	The timeout period for fetching a binary file during a hot upgrade.	120s	[1s,1200s]	false
`hot_upgrade_graceful_exit_timeout`	The timeout period for the OBProxy to cancel the hot upgrade.	120s	[0s, 30d]	false
`log_cleanup_interval`	The interval for cleaning up logs.	3600s	[5s,30d]	false
`log_dir_size_threshold`	The maximum size of the OBProxy logs. If this parameter value is exceeded, log cleanup is enabled.	64GB	[256MB,1T]	false
`conn_id_fetch_count`	The number of connection IDs you can obtain at a time.	10240	[10,100000]	false
`long_async_task_timeout`	The timeout period for asynchronous tasks that run on the OBServer	60s	[1s,1h]	false

name	info	default	range	need reboot
short_async_task_timeout	The timeout period for asynchronous tasks that may require short-term access to an OBServer	5s	[1s,1h]	false
client_max_connections	The maximum number of clients that can connect to the OBProxy.	8192	[0,65535]	false
observer_query_timeout_delta	The time period that is added to the ob_query_timeout value due to the network latency. ob_query_timeout indicates the timeout period for querying the OBServer.	20s	[1s,30s]	false
connect_observer_max_retries	The maximum number of retries when the OBProxy fails to connect to the OBServer.	3	[2,5]	false
net_config_poll_timeout	The timeout period for network connections.	10ms	[0,]	false
default_inactivity_timeout	The default timeout period for TCP connections.	180000s	[1s,30d]	false
sock_send_buffer_size_out	The size of the send socket buffer.	0	[0,8MB]	false
sock_recv_buffer_size_out	The size of the receive socket buffer.	0	[0,8MB]	false

name	info	default	range	need reboot
sock_option_flag_out	Specifies the socket flag parameter	0	[0,1]	false
sock_packet_mark_out	Specifies the socket mark parameter	0	[0,1]	false
sock_packet_tos_out	Specifies the socket tos parameter	0	[0,1]	false
server_tcp_init_cwnd	The initial size of the TCP congestion window when you create a connection xxx.	0	[0,64]	false
proxy_mem_limited	The maximum memory size that can be used for the OBProxy. If this parameter value is exceeded, the OBProxy automatically suspends.	800MB	[100MB,4GB]	false
enable_flow_control	Specifies whether to enable throttling in the tunnel.	false	• true • false	false
flow_high_water_mark	The upper limit of the traffic in the tunnel.	0	[0,16MB]	false
flow_low_water_mark	The lower limit of the traffic in the tunnel.	0	[0,16MB]	false

name	info	default	range	need reboot
flow_consumer_reenable_threshold	The maximum number of times that data is forwarded in the tunnel.	0	[0,131072]	false
flow_event_queue_threshold	The maximum number of tasks xxx	0	[0,20]	false
tunnel_request_size_threshold	The maximum request size that the OBProxy can receive. If this parameter value is exceeded, the OBProxy forwards client requests without parsing through the tunnel.	84KB	[4KB, 16MB]	false
default_buffer_size	The default block size for requests and responses.	8KB	[1KB, 64KB]	false
default_buffer_water_mark	The default water mark for the buffer.	32KB	[4B, 64KB]	false
enable_trans_detail_stats	Specifies whether collect transaction status information.	true	• true • false	true
enable_mysqlsm_info		true		false
enable_report_session_stats	Specifies whether to report session statistics to internal tables.	true	• true • false	false
slow_transaction_time_threshold	The maximum amount of time for executing a transaction. If this parameter value is exceeded, a warning log is printed.	5s	[0s,30d]	false

name	info	default	range	need reboot
`slow_proxy_process_time_threshold`	The maximum amount of time for the OBProxy to pre-process requests. If this parameter value is exceeded, a warning log is printed.	2ms	[0s,30d]	false
`query_digest_time_threshold`	The maximum amount of time for processing a request. If this parameter value is exceeded, a warning log is printed to obproxy_digest.log.	100ms	[0s,30d]	false
`slow_query_time_threshold`	The maximum amount of time for processing a slow query. If this parameter value is exceeded, a warning log is printed to obproxy_slow.log.	500ms	[0s,30d]	false
`congestion_failure_threshold`	The maximum number of server failures in a congestion_fail_window cycle. If this parameter value is exceeded, the OBProxy adds the server to the blacklist.	5	[0,]	false
`min_keep_congestion_interval`	The minimum amount of time that is waited to remove a server from the blacklist.	20s	[1s,1d]	false

name	info	default	range	need reboot
`congestion_fail_window`	The time period during which server failures are calculated. This parameter is used in conjunction with congestion_failure_threshold.	120s	[1s,1h]	false
`congestion_retry_interval`	The interval of retries for a live but unavailable server that is added to the blacklist	20s	[1s,1h]	false
`enable_congestion`	Specifies whether to enable the blacklist.	true	• true • false	false

Parameters that do not take effect until the OBProxy is restarted

name	info	default	range	need reboot
`stack_size`	The size of the thread stack. Specify this parameter when you create a thread.	1MB	[1MB,10MB]	true
`work_thread_num`	The number of worker threads. Specify this parameter when you initialize worker threads.	8	[1,128]	true
`task_thread_num`	The number of task threads. It is used when you initialize task threads	2	[1,4]	true
`dedup_queue_thread_num`	The number of threads to run fetch tasks	2	[1,4]	true

dedup_queue_bucket_num	`dedup_queue` The parameter `bucket_num` of	256	[256,1024]	true
dedup_queue_task_queue_size	`dedup_queue` The parameter `task_queue_size` of	1024	[512,1024]	true
kv_cache_bucket_num	It is used when you initialize kvcache	1024	[512,2048]	true
net_accept_threads	The number of threads to run the accept requests	2	[0,8]	true
frequent_accept	The parameter to initialize the net accept requests	true	• true • false	true

17.1.3.6. Backup and recovery

17.1.3.6.1. Overview

The backup and recovery feature of OceanBase Cloud Platform (OCP) supports full backup and incremental backup for ApsaraDB for OceanBase clusters and tenants. The log backup, full recovery, and incomplete recovery features are supported.

The following procedure shows how to back up and recover data in OCP:

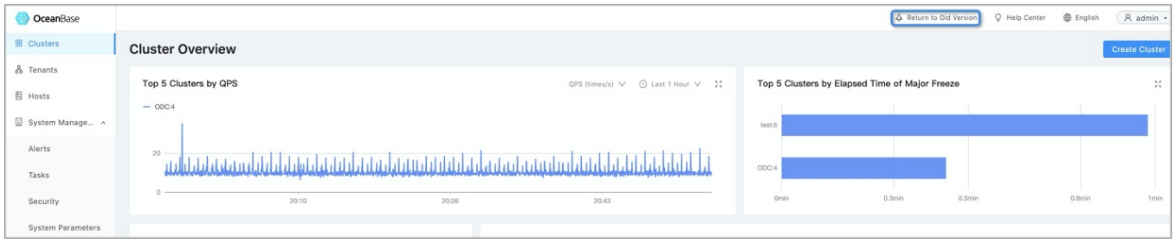
1. Create backup and recovery configuration files.
2. Upload and install backup and recovery agents.
3. Create a backup scheduling task for a cluster.
4. View the result of the backup task that is initiated by the scheduling task.
5. Initiate a recovery task.
6. Connect to the recovered tenant and view the recovered data.

17.1.3.6.2. Go to the Backup and Recovery page

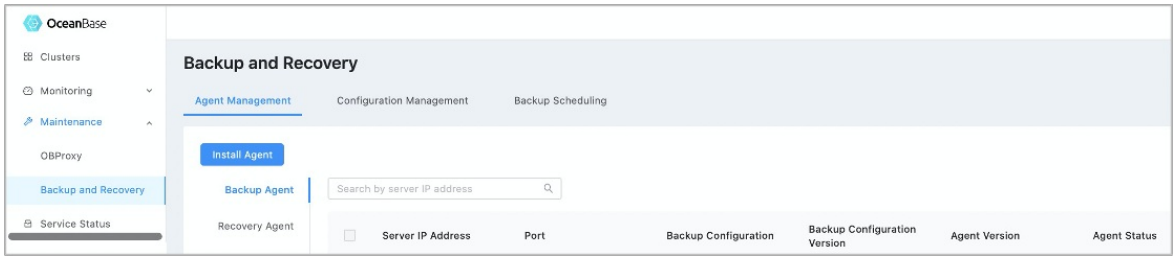
The Backup and Recovery page is displayed in the old version of OceanBase Cloud Platform (OCP). To use the backup and recovery features, you must switch the OCP version to the old version. On the Backup and Recovery page, you can perform O&M operations, use the backup scheduling feature, and view tasks.

Procedure

1. Log on to OCP.
2. In the upper-right corner of the page, click **Return to Old Version**.



3. In the left-side navigation pane, choose **Maintenance > Backup and Recovery**.



17.1.3.6.3. Preparations

Before you back up and recover data, you must prepare metadatabases and storage media.

Create a metadatabase for backup and recovery

When you install OCP based on Docker images, four versions of metadatabases are created by default within a metadatabase tenant. The versions are backup1472, backup147x, backup21, and backup2230. You can select to use these four versions based on your ApsaraDB for OceanBase version.

Prepare a storage medium

You can select Object Storage Service (OSS) or Network File System (NFS) as the storage medium for backup and recovery. NFS is applicable to independent external scenarios, and OSS is applicable to Apsara Stack scenarios.

OSS configuration

You must obtain the following settings to set the required parameters when you perform related operations on the Configuration Management tab:

- OSS account
- Bucket
- Endpoint
- AccessKey pair: An AccessKey pair is composed of an AccessKey ID and an AccessKey secret. The AccessKey pair is used to perform access identity verification.

NFS configuration

The backup agent, recovery agent, OceanBase Cloud Platform (OCP), and the ApsaraDB for OceanBase cluster to be restored must have access to the specified NFS directory for backup and recovery. Therefore, the NFS directory must be mounted to these hosts. The mounted on-premises directory must be the same as the directory in the backup configuration file.

- NFS server configuration

In this example, the directory name is obbackup.

```
yum install -y nfs-utils portmap
service nfs start
echo '/obbackup *(rw,all_squash,anonuid=500,anongid=500)' >/etc/exports
chmod 777 /obbackup
service nfs restart
exportfs
```

- NFS client configuration

The backup agent, recovery agent, OCP, and the ApsaraDB for OceanBase cluster to be restored must be mounted as a NFS client. Otherwise, data cannot be recovered.

```
showmount -e 11.166.84.52 #The IP address is an example IP address.
mkdir /obbackup
chmod 777 /obbackup
mount -o soft 11.166.84.52:/docker /obbackup
```

17.1.3.6.4. Manage backup and recovery configuration files

17.1.3.6.4.1. Add a backup configuration file

This topic describes how to add a backup configuration file.

Prerequisites

- A metadatabase for backup and recovery is created.
- A storage medium is prepared.

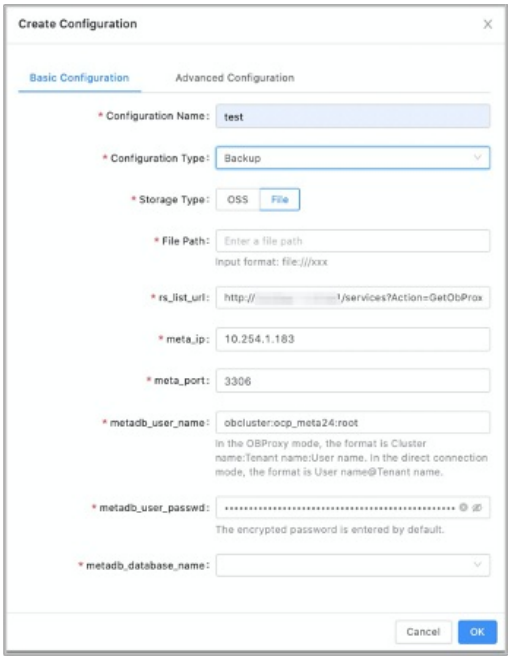
Procedure

1. Log on to OceanBase Cloud Platform (OCP).
2. Switch to the old version of OCP and open the Backup and Recovery page.
3. On the **Configuration Management** tab, click **Create Configuration** in the upper-left corner.
4. Set the required parameters on the **Basic Configuration** tab.
   i. Set the **Configuration Name** parameter.
   ii. Set **Configuration Type** to **Backup**.
   iii. Set the **Storage Type** parameter.

By default, the database information of the OCP metadatabase is automatically filled in. The OCP database includes a backup and recovery metadatabase. Modify the settings based on the actual metadatabase.

- If you set Storage type to **OSS**, you must set the File Path, access_key_id, secret_access_key, rs_list_url, meta_ip, meta_port, metadb_user_name, metadb_user_passwd, and metadb_database_name parameters. Some parameters are automatically set, confirm whether the settings are correct.
- If you set Storage type to **File**, you only need to enter the related information of the mounted NFS file.

Parameter	Description
File path	The OSS file path or the file path where the NFS file is mounted.
host	The endpoint.
access_key_id	The AccessKey ID of the OSS account.
secret_access_key	The AccessKey secret of the OSS account.
rs_list_url	The RS List URL provided by current OCP is filled by default.
meta_ip	The IP address of the backup and recovery metadatabase.
meta_port	The port number of the backup and recovery metadatabase.
metadb_user_name	The username of the backup and recovery metadatabase.
metadb_user_passwd	The password of the backup and recovery metadatabase.
metadb_database_name	Select an appropriate metadatabase as prompted.



5. (Optional) On the **Advanced Configuration** tab, set the required parameters of **Advanced Configuration**. The parameters of Advanced Configuration are automatically set. You do not need to modify the related settings.
6. Click **OK**.

17.1.3.6.4.2. Add a recovery configuration file

This topic describes how to add a recovery configuration file.

Prerequisites

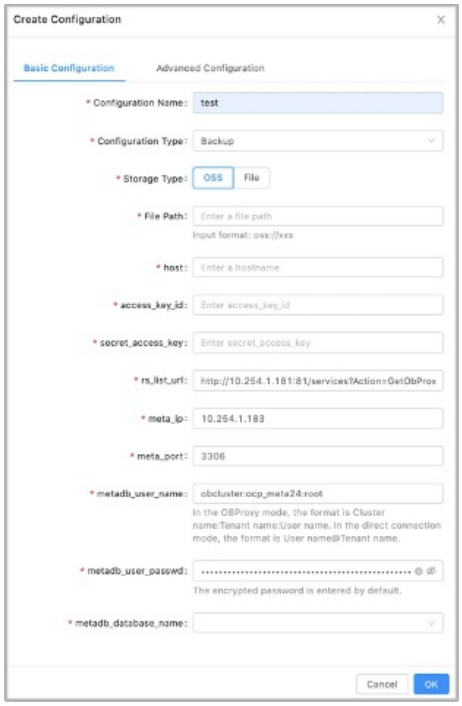
- A metadatabase for backup and recovery is created.
- A storage medium is prepared.

Procedure

1. Log on to OceanBase Cloud Platform (OCP).
2. Switch to the old version of OCP and open the Backup and Recovery page.
3. On the **Configuration Management** tab, click **Create Configuration** in the upper-left corner.
4. Set the required parameters on the **Basic Configuration** tab.
   i. Set the **Configuration Name** parameter.
   ii. Set **Configuration Type** to **Recovery**.
   iii. Set the required parameters.

By default, the database information of the OCP metadatabase is automatically filled in. The OCP database includes a backup and recovery metadatabase. Modify the settings based on the actual metadatabase.

Parameter	Description
ocpMetaDb.clusterName	The name of the cluster where the backup and recovery metadatabase resides. The cluster must be managed by OCP.
ocpMetaDb.tenantName	The tenant to which the backup and recovery metadatabase belongs.
ocpMetaDb.userName	The username of the backup and recovery metadatabase.
ocpMetaDb.dbName	Select an appropriate metadatabase as prompted.
ocpMetaDb.encryptedPassword	The password of the backup and recovery metadatabase.
ocpMetaDb.configUrl	The configUrl of the cluster where the backup and recovery metadatabase resides. This parameter is automatically set and you do not need to modify the related settings.



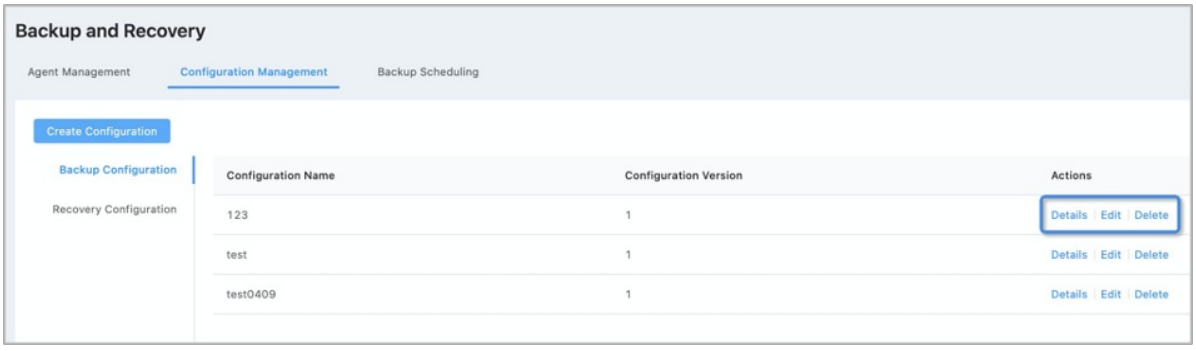
5. (Optional) On the **Advanced Configuration** tab, set the required parameters of **Advanced Configuration**. The parameters of Advanced Configuration are automatically set. You do not need to modify the related settings.
6. Click **OK**.

17.1.3.6.4.3. Manage a configuration file

The related parameters in backup and recovery configuration files must be correctly set. Otherwise, the related backup and recovery agents cannot work as required. This results in failed backup and recovery.

You can manage your configuration files on the Configuration Management tab based on your business requirements. You can perform the following operations on a configuration file:

- **View the details of the configuration file.**
- **Modify the configurations.**
- **Delete a backup or recovery configuration file.**



17.1.3.6.5. Install backup and recovery agents

17.1.3.6.5.1. Install backup and recovery agents

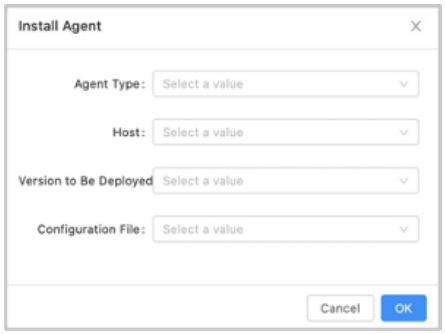
This topic describes how to install backup and recovery agents.

Prerequisites

A host is required to install a backup or recovery agent. Make sure that an idle host is available. If no available hosts are available, you can add a host on the **Host Management** page of the new version of OceanBase Cloud Platform (OCP) based on the related documentation of **Host Management**.

Procedure

1. Log on to OCP.
2. Switch to the old version of OCP and open the Backup and Recovery page.
3. On the **Agent Management** tab, click **Install Agent**.
4. Set the **Agent Type** parameter and specify a host.
5. Set the **Version to Be Deployed** parameter. If no versions are available, you can click Add Version to upload a new installation package.
6. The related backup or recovery configuration file is automatically filtered based on the specified deployment version. If no configuration files are available, add a backup configuration file first.
7. Click OK.



8. After you click OK, a dialog box appears on the page. You can click the link to view the progress.

The progress page is displayed in the new version of OCP. After you view the progress, you need to switch back to the old version. Then, you can perform related O&M operations on the agent.

The process of installing a recovery agent is the same as that of the backup agent. Both backup and recovery agents can be installed on the same host. However, a version is based on the version number of the agent installed first. Different versions cannot be separately specified.

17.1.3.6.5.2. Manage backup and recovery agents

Installed backup and recovery agents are displayed on the Agent Management tab of the Backup and Recovery page. You can perform some common O&M operations on these installed agents.

You can perform the following operations:

- View the task history of an agent. In the **Task History** column, click **View** to view the historical O&M status of the agent.
- Restart an agent.
- Delete an agent.
- Upgrade an agent.
- Update the configurations of an agent.

  If you want to update the configuration file of an agent, you can modify the configurations on the **Configuration Management** tab, and then update the configurations. The system automatically pushes the latest configurations to the agent side.

17.1.3.6.6. Run backup tasks

17.1.3.6.6.1. Cluster backup scheduling

After you have created backup and restoration configuration profiles and installed the backup component, you can back up data at any time.

Create a cluster backup scheduling task

Prerequisites

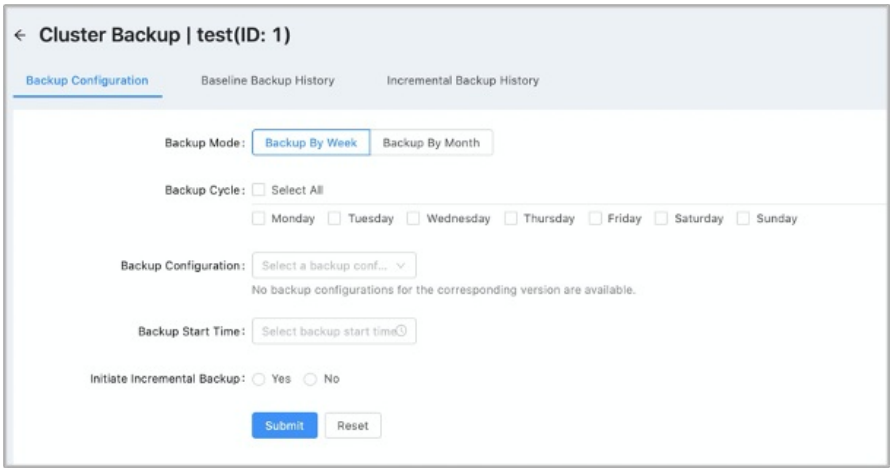
- You have created the backup configuration profile of the corresponding version.
- You have installed the backup component.

Procedure

1. Log on to the OceanBase Cloud Platform (OCP) console.
2. In the upper-right corner, click **Return to Old Version**.
3. In the left-side navigation pane, choose **Clusters**.
4. In the **Actions** column of the specific cluster, click **Cluster Backup**.
5. Specify related configuration parameters.
   i. Set the backup mode and backup period.
   - The backup mode includes weekly backup and monthly backup.
   - The backup period varies with the backup mode. You can select the weekday or the day of a month for backup.

   ii. Select the backup configuration profile.
   - The drop-down list only displays profiles that match the specified version.
   - If no matching backup configuration profile is available for the current version, you must create one first.

   iii. Set the backup start time.
   iv. Specify whether to enable incremental backup, where incremental backup is to back up logs.



6. Click **Submit**.

Manage cluster backup scheduling

After you have created a cluster backup scheduling task, the system automatically initiates the backup task at the specified time and period. You can also modify the task based on business needs. The system supports the following management operations at present:

- Pause a backup scheduling task.

In the Actions column of the specific backup task, choose Actions > Pause to temporarily suspend the task.

- Restart a backup scheduling task.

In the Actions column of the specific backup task, choose Actions > Restart to restart the task. The system will initiate the task at the specified time and period.

View backup history

On the **Backup Scheduling** tab of the **Backup and Recovery** page, you can view the basic information of all the scheduling tasks, including the cluster name, tenant name, backup mode, backup period, status of the current scheduling task, status of the baseline backup task, incremental backup task, and initiator.

You can click the name of a cluster to view its backup scheduling history, baseline backup history, incremental backup history, and restoration history. On the Backup Scheduling History tab, you can filter backup scheduling tasks by state.

- Backup Scheduling History: This tab shows the following information: the task name, ID, cluster, initiator, status, progress, start time, and end time.
- **Baseline Backup History**: This tab shows the following information: the task ID, backup type, task type, task quantity, cluster, tenant, start time, end time, data version, backup URL, backup status, and backup size. You can select a backup task to initiate a data restoration.
- **Incremental Backup History**: This tab shows the following information: the cluster name, tenant whitelist, majorVersion, backup URL, agentServerIp, checkpoint, delay (in seconds), status, stop flag, and error.

17.1.3.6.6.2. Tenant backup scheduling

You cannot back up a tenant and the related cluster at the same time.

Only one tenant backup schedule can be created in a cluster.

- If no backup schedules are available, you can create a cluster backup schedule or a tenant backup schedule.
- If a cluster backup schedule is created, all tenants in the cluster follow the scheduling configuration of the cluster by default. You cannot configure a tenant scheduling task.
- If a tenant backup schedule is created, you cannot create a backup schedule for the related cluster. Before you configure a cluster backup schedule, you must cancel the backup scheduling task of the related tenant and suspend the related incremental backup task.

Prerequisites

No running backup scheduling tasks or incremental backup tasks in the cluster to which the specified tenant belongs.

Procedure

1. Log on to OceanBase Cloud Platform (OCP).
2. In the upper-right corner of the page, click **Return to Old Version**.
3. In the left-side navigation pane, click **Tenants**.
4. Click **Cluster Backup** in the **Actions** column of the specified cluster.

5. Set the required parameters.
   i. Set the **Backup Mode** and **Backup Cycle** parameters.
   - The valid values of the **Backup Mode** parameter are **Backup By Week** and **Backup By Month**.
   - The backup cycle can be set to the specified day or day of the week for backup based on the specified backup mode.

   ii. Set the **Backup Configuration** parameter.
   - Only the backup configuration files of the related version are displayed in the drop-down list.
   - If no backup configuration files are available, create a backup configuration file before you configure a cluster backup schedule.

   iii. Set the **Back Start Time** parameter.

   iv. Set the **Initiate Incremental Backup** parameter. Incremental backup is log backup.
6. Click **Submit**.

Manage a tenant backup schedule

After you create a tenant backup schedule, the system automatically initiates backup tasks based on the preset cycle and time. You can modify the backup scheduling tasks based on your business requirements. The following operations are supported to manage your schedules:

- Suspend a backup scheduling task. Click **Suspend** in the Actions column of a specified backup scheduling task to suspend the task.
- Click **Rerun** in the Actions column of a specified backup scheduling task to rerun the task. Then, the system automatically initiates backup tasks based on the preset cycle and time.

17.1.3.6.6.3. Back up now

You can initiate an instant backup task for a cluster or tenant that has a backup schedule. If a backup schedule is available, you can manually perform a baseline backup task.

> **Note**
>
> A major freeze version of ApsaraDB for OceanBase can only have one baseline backup task. If multiple backup tasks are initiated based on the same version, errors are reported for subsequent backup tasks. However, the backup tasks of this version are available.

Prerequisites

A backup scheduling task is configured for a cluster or a tenant.

Procedure

1. Log on to OceanBase Cloud Platform (OCP).
2. In the upper-right corner of the page, click **Return to Old Version**.
3. In the left-side navigation pane, click **Tenants** or **Clusters**.
4. Click **Back Up Now** in the **Actions** column of the tenant or cluster.

5. In the dialog box that appears, click **OK**.

17.1.3.6.7. Recover backup data

After the system performs a backup operation, it can initiate a recovery task based on the backup. The minimum granularity of recovery tasks is tenant-level data. Multiple tenants can be recovered in batches.

Prerequisites

A baseline backup task is performed on the cluster or tenant to be recovered.

Procedure

1. Log on to OceanBase Cloud Platform (OCP).
2. Go to the Backup and Recovery page.
3. On the **Backup Scheduling** tab, click the name of the specified cluster.
4. On the **Baseline Backup History** tab, find the baseline backup to be restored.
5. In the **Actions** column, click **Initiate Recovery**. The click position and the recovery version have no relations. The system automatically calculates and chooses a baseline version for recovery based on recovery time.
6. Specify **Cluster for Recovery** and **Restore to Point in Time**. The cluster to be recovered can be a source cluster or another cluster.
7. Enter the user password.
8. Select the tenant to be restored and confirm the settings of **Target Tenant Name, Restore Locality, Restore Pool List, and Restore Primary Zone**. The system automatically fills the related fields based on the backup information. You must confirm whether the filled content is correct.
   - If you recover data to the source cluster, you must change the tenant name to a new name.
   - Restore Pool List indicates the related resource configurations of the tenant. The resource pool name cannot exist in the destination cluster.
9. Click **OK**. After the recovery request is successfully initiated, a success message appears.

View the status of a recovery task

In the **Recovery History** column, you can view the status of the recovery task. You must refresh the page to view the updated information.

The recovery process consists of the following two phases. Only when both phases are successful, the tenant recovery is successful.

- Baseline recovery
- Incremental recovery

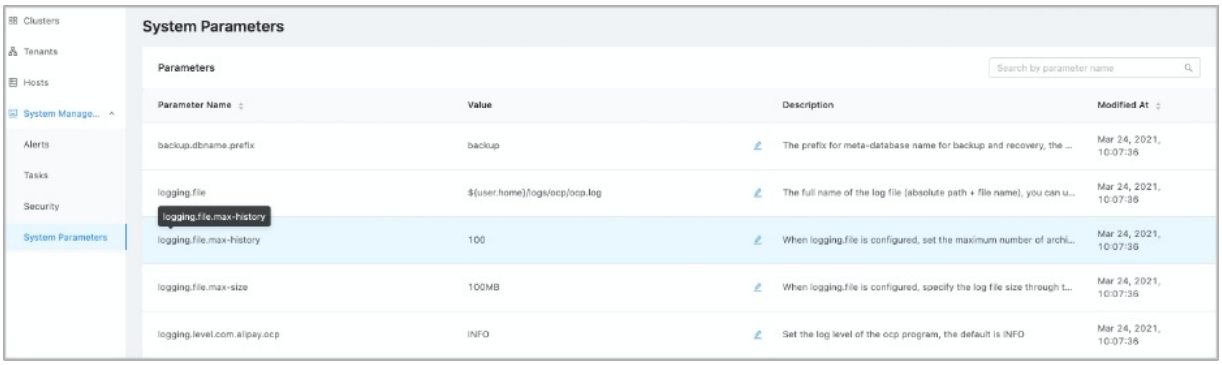
Use a recovery tenant

The connection password of the recovery tenant is empty. When the message is prompted, enter to confirm.

17.1.3.6.8. Monitoring and alert configurations

Backup and recovery settings are configured as system parameters in OceanBase Cloud Platform (OCP).

On the System Management>System Parameters page of the new version of OCP, enter backup in the search box to query and view the parameters of backup and recovery.



Monitoring configurations

In the current OCP version, you can manage and configure the following parameters of backup and recovery:

- Status monitoring of agents.
- Monitoring of failed baseline backup tasks.
- Monitoring of incremental backup task latency. For more information, see `ocp.backup.alarm.inc-backup-delay-threshold`.
- Monitoring of failed backup scheduling tasks.
- Monitoring of expired backup file cleaning. For more information, see `ocp.backup.alarm.backup-data-retention-days`.
- Monitoring for the log cleaning of backup agents. For more information, see `ocp.backup.alarm.backup-liboblog-expire-days`.

> **Note**
>
> The current version does not support the monitoring for the capacity of backup files.

Alert configurations

The following table describes the parameters of alerts.

Parameter	Default value	Description
ocp.backup.alarm.base-backup-last-finished-threshold	12960	The baseline has not been successfully backed up for 9 days, the initial threshold is 9 days (12960 minutes), the unit is minutes
ocp.backup.alarm.base-backup-timeout	10	Baseline backup scheduling timeout period (minutes)
ocp.backup.alarm.inc-backup-delay-threshold	3600	Incremental backup delay alarm threshold (seconds)

Parameter	Default value	Description
ocp.backup.alarm.last-data-backup-max-interval-minutes	1440	Check whether the baseline backup task failed in a certain time range recently, the default is 1 day, the unit is minute; that is, the maximum allowable interval between the last successful data backup and the current time

For more information about alert configurations, see related documentation to subscribe and configure.

17.1.3.7. Manage users and permissions

17.1.3.7.1. Overview of access control

In the OCP platform, users are those who use the OCP platform and operate functional systems. A role is a set of permissions and functions as a carrier of the permissions. Each user must assume one or more roles and have the permissions of these roles before the user performs the corresponding operations in the OCP platform. The permissions of a user are the union of the permissions of all the roles that the user assumes.

The initial account admin in the OCP platform is a user that has the system administrator role. The admin account has management permissions on all the clusters and functional modules in the OCP platform. A system administrator can create another user and assign one or more roles to the user to allow the user to assume corresponding functions in the business architecture. For example, the system administrator can create a user that has the user management permission USER_MANAGER so that the user can create other users on the behalf of the administrator.

17.1.3.7.2. Manage users

17.1.3.7.2.1. Create a user

For security purposes, such as information security and traditional security, we recommend that you grant the minimum permissions that are required by a user when you create the user. The minimum permission set of OCP is the PROFILE role. A user who has this role can perform operations, such as accessing the personal center and modifying personal information and the password box.

OCP initially presets common roles. These common roles are called default roles. The default roles include roles, such as the system administrator role ADMIN that has all management permissions and the TENANT_MANAGER role that has all the management permissions on ApsaraDB for OceanBase tenants. The default roles basically cover general requirements for managing user permissions. If you need to provide finer-grained permissions on clusters and tenants through a user, you can create a custom role and specify the specific cluster ID and tenant ID when you create the user.

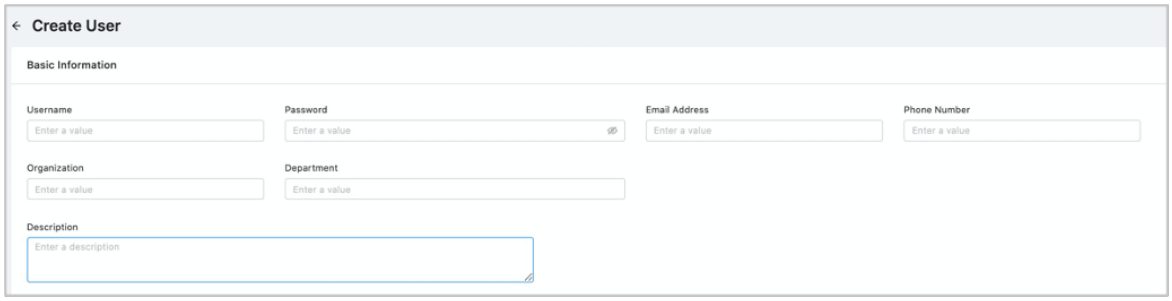
Prerequisites

The current logon user has the system administrator role ADMIN or the user management role USER_MANAGER and the corresponding permissions.

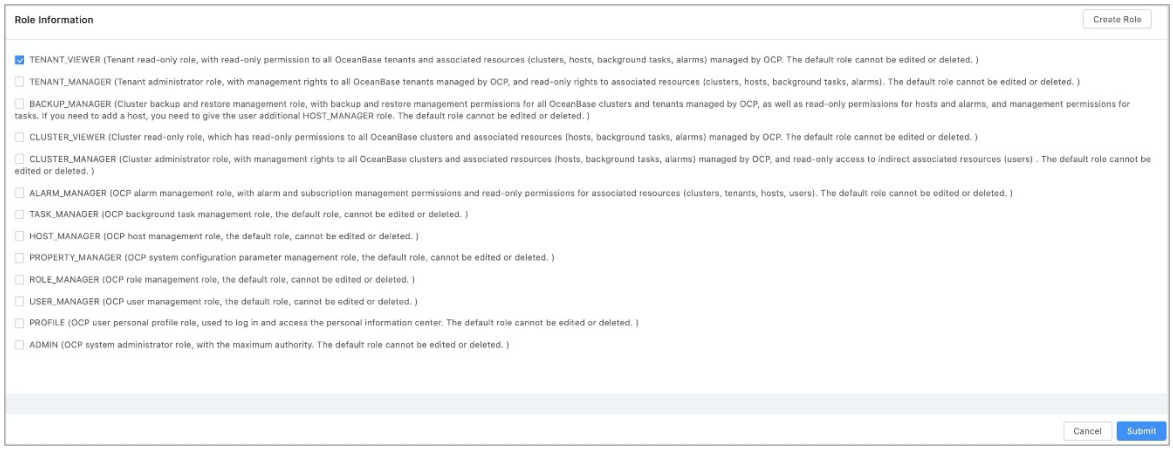
Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Security**.
3. In the upper-right corner of the page, click **Create User**.

4. Specify basic information for the new user.
   - Specify **Username, Password, Email Address, Phone Number, Organization, and Department** for the new user, and enter role descriptions.
   - Among the preceding parameters, Username, Password, and Email Address cannot be left blank.
   - The username must start with a lowercase letter, and can contain lowercase letters, digits, underscores (_), hyphens (-), and periods (.).The username must be 4 to 48 characters in length. The username cannot be modified after it is created.
   - The password must be 8 to 32 characters in length, and must contain at least two digits, two uppercase letters, two lowercase letters, and two special characters (_+@#$%).



5. Specify role information for the new user.
   - One or more roles must be specified for the user.
   - If an appropriate role does not exist in OCP, you can click **Create Role** in the upper-right corner of the section to create a role, and assign the role to the new user.



6. In the lower-right corner of the page, click **Submit**.

17.1.3.7.2.2. View a user list

System administrators or user administrators can view all the user information in the current OCP, including the email address, phone number, department, organization, and description of the user and the roles that are owned by the user.

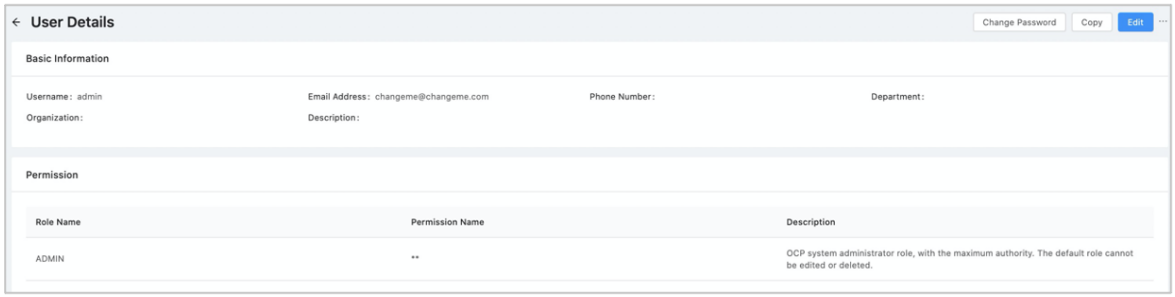
Prerequisites

The current logon user has the system administrator role ADMIN or the user management role USER_MANAGER and the corresponding permissions.

Procedure

1. Log on to OCP.

2. In the left-side navigation pane, choose **System Management > Security**.
3. In the search box, enter a username or a keyword of the username to search for the specified user.
4. In the search results, click the username to go to the **User Details** page. On the **User Details** page, you can view the email address, phone number, department, organization, and description of the user and the roles that are owned by the user.



17.1.3.7.2.3. View user details

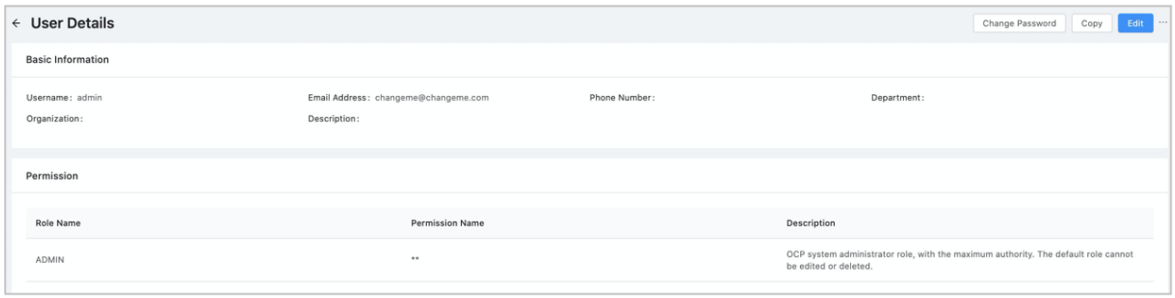
System administrators or user administrators can view all the user information in the current OCP, including the email address, phone number, department, organization, and description of the user and the roles that are owned by the user.

Prerequisites

The current logon user has the system administrator role ADMIN or the user management role USER_MANAGER and the corresponding permissions.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Security**.
3. In the search box, enter a username or a keyword of the username to search for the specified user.
4. In the search results, click the username to go to the **User Details** page. On the **User Details** page, you can view the email address, phone number, department, organization, and description of the user and the roles that are owned by the user.



17.1.3.7.2.4. Copy a user

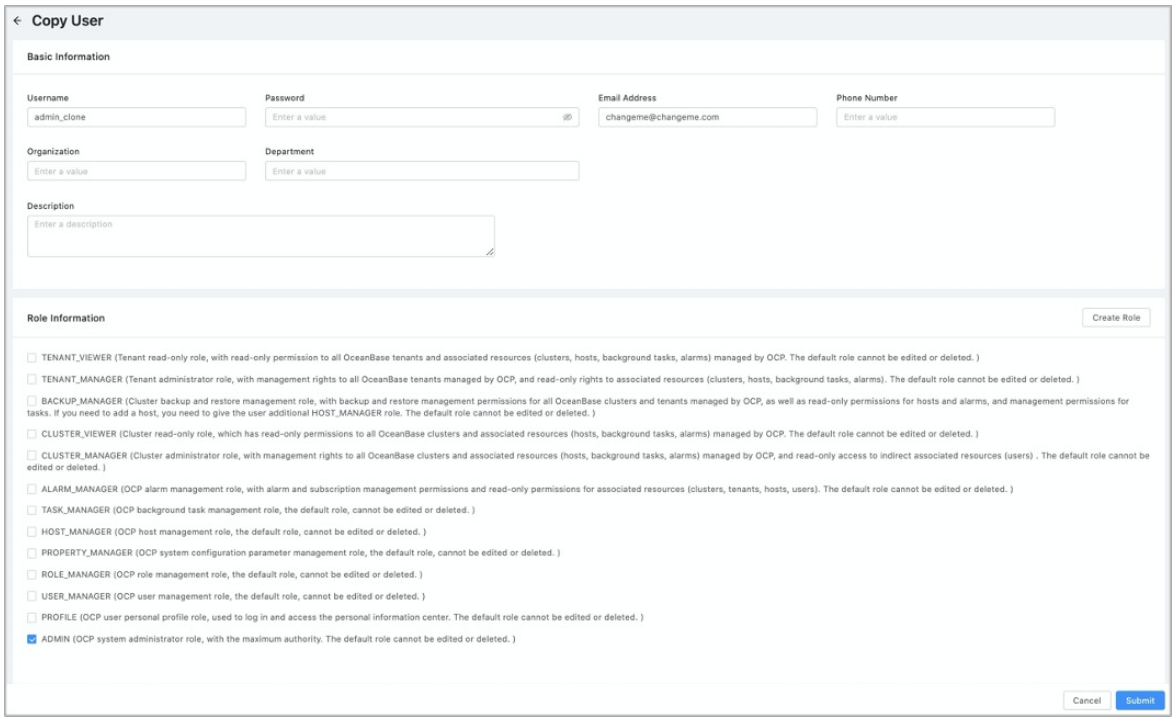
If you need to create a user whose basic information is the same as that of an existing user, you can copy the existing user to create the user. The user copying method eliminates the need of repeatedly entering same basic information, and applies to scenarios where you need to create users in batches.

Prerequisites

The current logon user has the system administrator role ADMIN or the user management role USER_MANAGER and the corresponding permissions.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Security**.
3. Find the user to be copied, and click **Copy** in the corresponding **Actions** column.
4. Specify basic information and role information for the new user. The basic information and role of the new user are the same as those of the user to be copied except for the username and the password. You can edit information based on the information of the user to be copied.



5. In the lower-right corner of the page, click **Submit**.

17.1.3.7.2.5. Edit a user

After users are created, you can modify the basic information and permission information about the users, such as passwords, email addresses, and phone numbers. After you modify the role of a user, the modifications take effect the next time the user logs on.

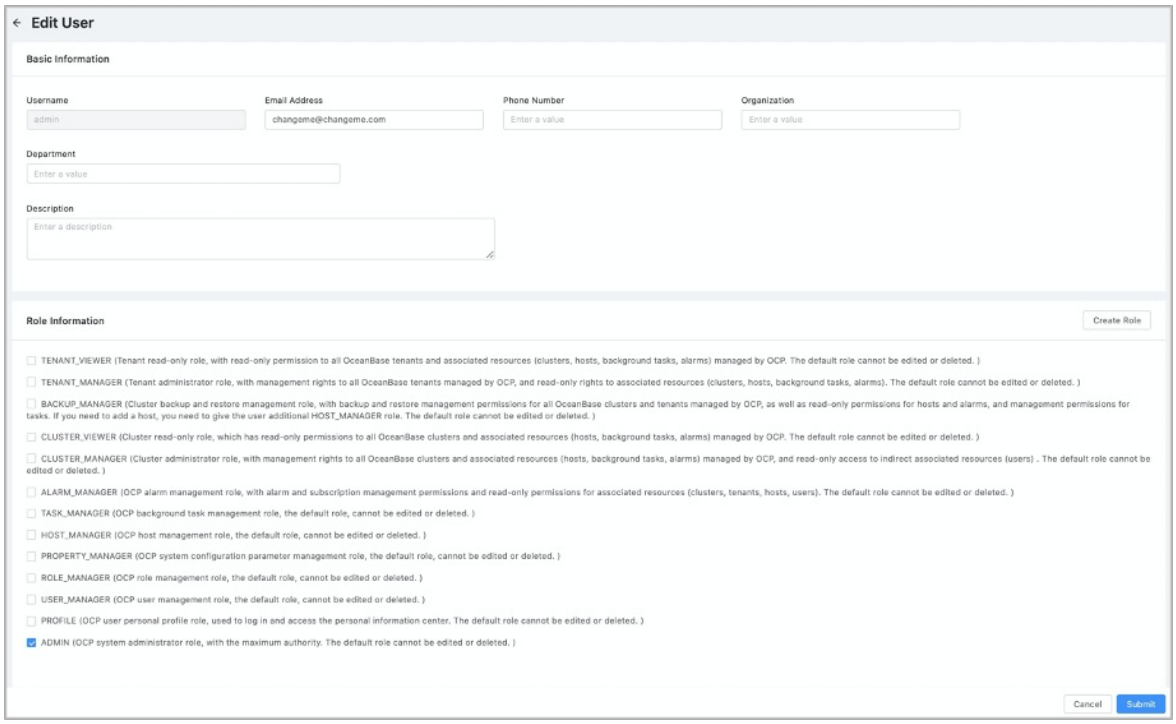
Prerequisites

The current logon user has the system administrator role ADMIN or the user management role USER_MANAGER and the corresponding permissions.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Security**.
3. Find the specified user, and click **Edit** in the corresponding **Actions** column.

4. Enter basic information about the user again. You can also reassign a role to the user.



5. In the lower-right corner of the page, click **Submit**.

17.1.3.7.2.6. Delete a user

User information involves in recording and tracking the historical operations of users. Therefore, after a user is deleted in OCP, the account information, such as the username and the email address, is retained. If you create a user that has the same username or email address as the deleted user, the system prompts you that a user that has the same username or email address already exists. You can select another username and email address as prompted and try again.

Prerequisites

The current logon user has the system administrator role ADMIN or the user management role USER_MANAGER and the corresponding permissions.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Security**.
3. Find the specified user, expand the hidden list in the corresponding **Actions** column, and click **Delete**.

4. In the dialog box that appears, click **Delete**.

17.1.3.7.3. Manage roles

17.1.3.7.3.1. Overview

A role in OCP is a set of permissions. Roles only take effect after they are assigned to users. OCP roles consist of default roles and custom roles.

OCP roles consist of default roles and custom roles.

- Default roles are preset roles that are built in the system. The default roles basically cover general requirements for managing user permissions. For example, the system administrator role ADMIN has all management permissions, and the TENANT_MANAGER role has all the management permissions on ApsaraDB for OceanBase tenants.
- Custom roles are the roles that are created by users in OCP. The custom roles have finer-grained management permissions on clusters and tenants than the initial default roles. For example, the specified users can view only the specified clusters and the specified tenants in the specified clusters. However, you can grant custom roles only the relevant management permissions on clusters, tenants, and backup and recovery and the system permissions to create clusters.

Default roles

Default role cannot be edited or deleted. The following table describes the default roles that are provided by OCP and the permissions of these roles.

Role name	Description	Permission list
TENANT_VIEWER	The tenant read-only role. This role has read-only permissions on all the ApsaraDB for OceanBase tenants that are managed by OCP and the associated resources (clusters, hosts, background tasks, and alerts).	CLUSTER:*:TENANT:*:READ, CLUSTER:*:READ, HOST:*:READ, TASK:*:READ, ALARM:*:READ
TENANT_MANAGER	The tenant administrator role. This role has management permissions on all the ApsaraDB for OceanBase tenants that are managed by OCP and read-only permissions on the associated resources (clusters, hosts, background tasks, and alerts).	CLUSTER:*:TENANT:*:*, CLUSTER:*:READ, HOST:*:READ, TASK:*:READ, ALARM:*:READ

Role name	Description	Permission list
BACKUP_MANAGER	The management role of cluster backup and recovery. This role has management permissions on the backup and recovery of all the ApsaraDB for OceanBase clusters and tenants that are managed by OCP. This role also has read-only permissions on hosts and alerts, and management permissions on tasks. If you need to add a host, you must grant an additional HOST_MANAGER role to the user.	CLUSTER:*:BACKUP:*:*, CLUSTER:*:READ, CLUSTER:*:TENANT:*:READ, HOST:*:READ, ALARM:*:READ, TASK:*:*
CLUSTER_VIEWER	The cluster read-only role. This role has read-only permissions on all the ApsaraDB for OceanBase clusters that are managed by OCP and the associated resources (hosts, background tasks, and alerts).	CLUSTER:*:READ, HOST:*:READ, TASK:*:READ, ALARM:*:READ
CLUSTER_MANAGER	The cluster administrator role. This role has management permissions on all the ApsaraDB for OceanBase clusters that are managed by OCP and the associated resources (hosts, background tasks, and alerts). This role also has read-only permissions on the indirectly associated resources (users).	CLUSTER:*:*, HOST:*:*, TASK:*:*, ALARM:*:*, USER:*:READ
ALARM_MANAGER	The alert management role in OCP. This role has management permissions on alerts and subscriptions and read-only permissions on the associated resources (clusters, tenants, hosts, and users).	ALARM:*:*, CLUSTER:*:READ, HOST:*:READ, USER:*:READ
TASK_MANAGER	The management role of background tasks in OCP.	TASK:*:*
HOST_MANAGER	The host management role in OCP. This role has all the management permissions on hosts.	HOST:*:*

Role name	Description	Permission list
PROPERTY_MANAGER	The management role of system configuration parameters in OCP.	PROPERTY:*:*
ROLE_MANAGER	The management role of roles in OCP. This role has all the role-related management permissions.	ROLE:*:*
USER_MANAGER	The user management role in OCP. This role has all the user-related management permissions.	USER:*:*
PROFILE	The user personal information role in OCP. This role is used to log on to and access the personal center.	PROFILE:*:*
ADMIN	The system administrator role in OCP. This role has the maximum permission of OCP.	**

Custom roles

You can grant custom roles only the relevant management permissions on clusters, tenants, and backup and recovery and the system permissions to create clusters. You can use custom roles and default roles in combination to provide a variety of authorization policies.

You can grant the following permissions to custom roles.

Permission type	Permission content
Cluster permissions	The permissions of the specified cluster: **Create Tenant**, **Read-only**, **Update**, **Delete**, or **All Permissions**.
Tenant permissions	The permissions of the **specified ApsaraDB for OceanBase tenant** in the **specified ApsaraDB for OceanBase cluster**: **Read-only**, **Update**, **Delete**, or **All Permissions**.
Backup and recovery permissions	The following permissions of the specified cluster: • **Create**: Install components and initiate backup and recovery. • **Read-only**: View backup and configuration status. • **Update**: Update configurations and maintain components. • **Delete**: Delete configurations.

Permission type	Permission content
System permissions	Permissions to **create ApsaraDB for OceanBase clusters**.

17.1.3.7.3.2. Create a role

If the preset default roles in OCP cannot meet your requirements, system administrators or role administrators can create custom roles and grant more fine-grained management permissions to the custom roles. You can use custom roles and default roles in combination to provide a variety of authorization policies.

> **Note**
>
> You can grant custom roles only the relevant management permissions on clusters, tenants, and backup and recovery and the system permissions to create clusters.

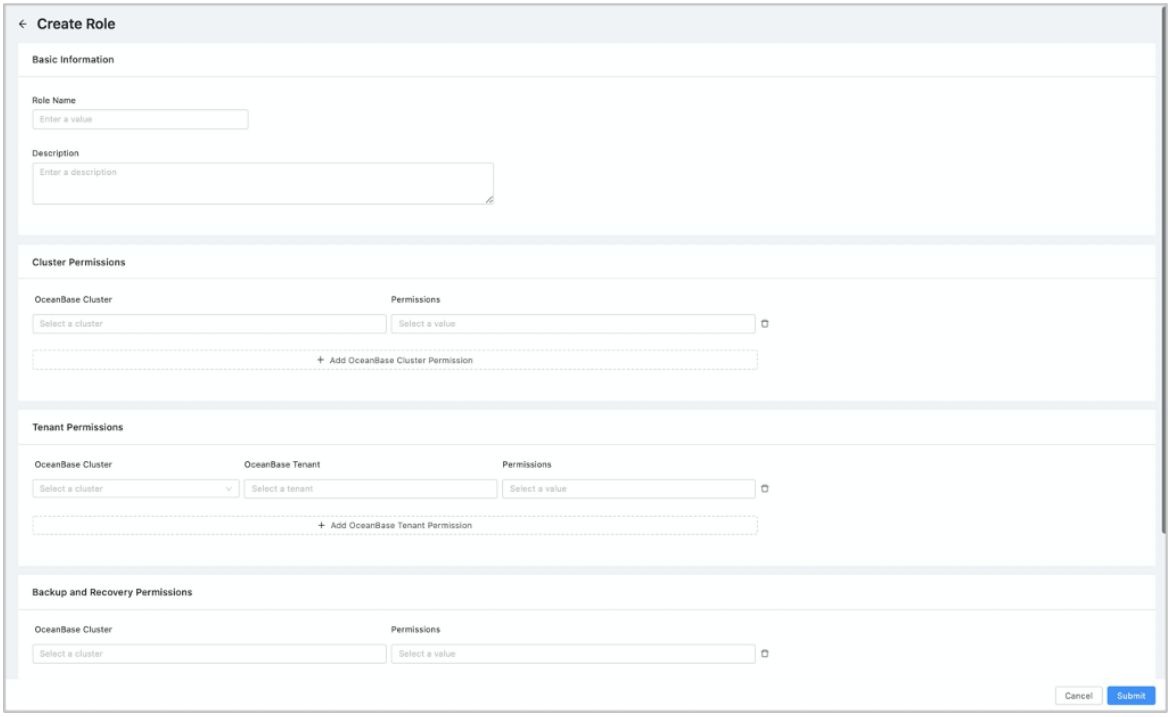
Prerequisites

The current logon user has the system administrator role ADMIN or the role management role ROLE_MANAGER and the corresponding permissions.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Security**.
3. In the upper-right corner of the **Role Management** tab, click **Create Role**.
4. Specify the **Role Name** and **Description** parameters.
5. Grant the role the relevant management permissions on clusters, tenants, and backup and recovery and the system permissions to create clusters based on service requirements.
   - No intersection exists between read-only permissions and management permissions, such as creation and editing permissions. If you need to grant management permissions to a role, you must also grant read-only permissions to the role.

- If tenant-relevant read-only or management permissions are required, you must also grant the role the read-only or management permissions on the cluster that corresponds to the tenant.



6. In the lower-right corner of the page, click **Submit**.

17.1.3.7.3.3. View role information

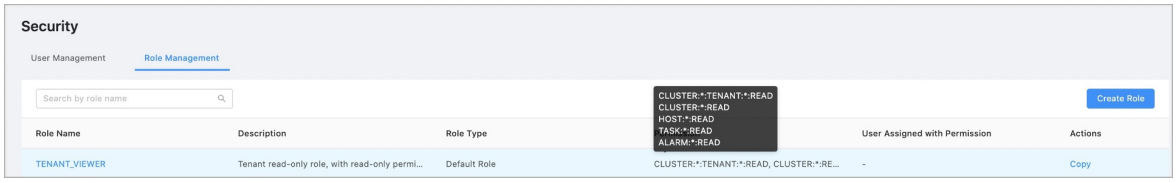
System administrators and role administrators can view information about specific roles or all the roles in the current OCP, including the role name, the description, and the permission content.

Prerequisites

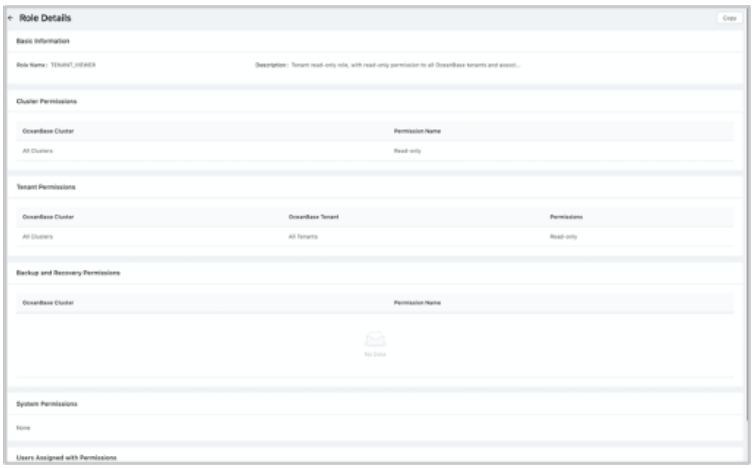
The current logon user has the system administrator role ADMIN or the role management role ROLE_MANAGER and the corresponding permissions.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Security**.
3. In the search box on the **Role Management** tab, enter the specified role name or a keyword, and click the search icon. In the list, you can view the following information about the specified role: **Description**, **Role Type**, **Permission** content, and **User Assigned with Permission**.



4. Click the role name in the search results. On the Role Details page, all the information about this role is displayed.



17.1.3.7.3.4. Copy a role

If you need to add other fine-grained permissions based on the existing role, you can create a role by copying a role.

Even if you create a role by copying a default role, the new role is still a custom role. You can grant the new role only the relevant management permissions on clusters, tenants, and backup and recovery and the system permissions to create clusters. The new role does not have the relevant system management features of the default role.

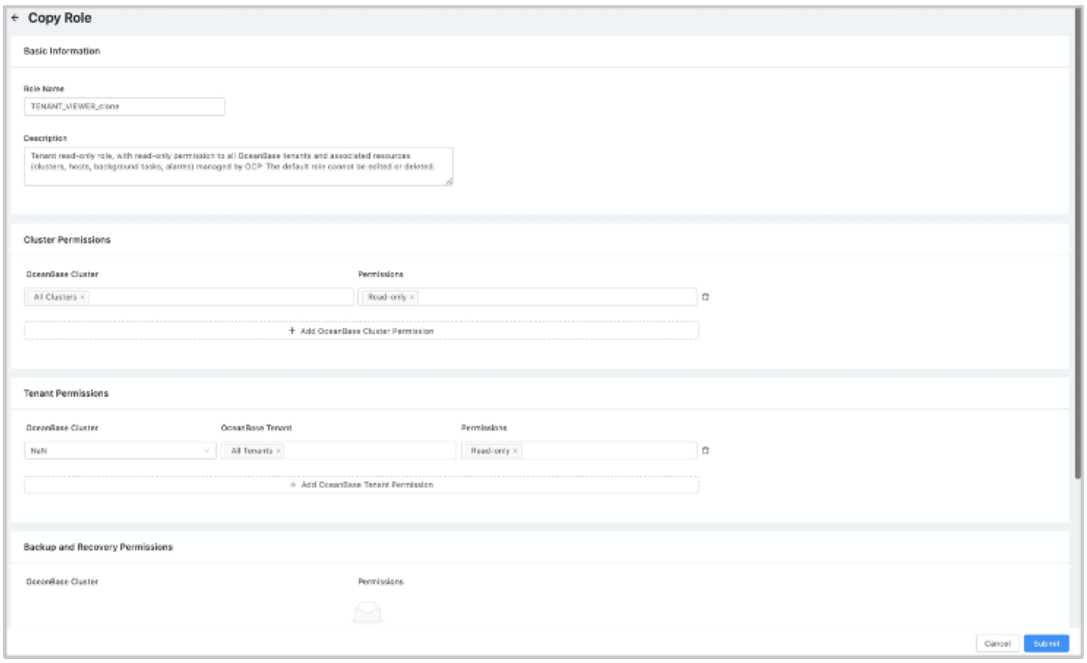
Prerequisites

The current logon user has the system administrator role ADMIN or the role management role ROLE_MANAGER and the corresponding permissions.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Security**.
3. On the **Role Management** tab, find the role to be copied, and click **Copy** in the corresponding **Actions** column.

4. Edit basic information and permission information about the new role.



5. In the lower-right corner of the page, click **Submit**.

17.1.3.7.3.5. Edit a role

After system administrators or role administrators create custom roles, they can modify the information and permissions of the custom roles anytime.

> **Note**
>
> - After you modify a role that has been assigned to a user, the modifications take effect the next time the user logs on.
> - Default roles cannot be modified.

Prerequisites

The current logon user has the system administrator role ADMIN or the role management role ROLE_MANAGER and the corresponding permissions.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Security**.
3. On the **Role Management** tab, find the specified role, and click **Edit** in the corresponding **Actions** column.
4. Modify the descriptions or permissions of the role based on your business requirements.
5. In the lower-right corner of the page, click **Submit**.

17.1.3.7.3.6. Delete a role

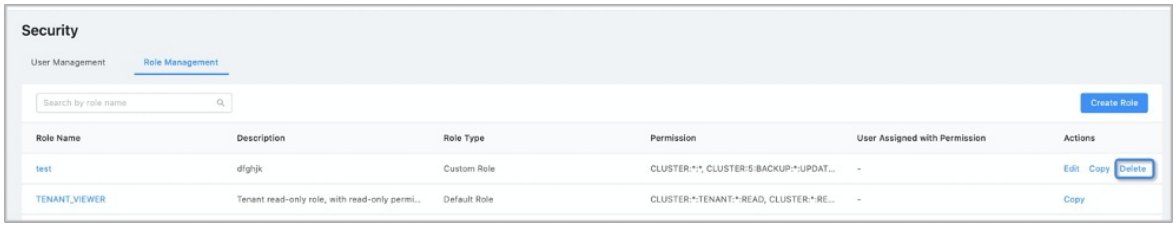
System administrators and role administrators can delete custom roles anytime based on business requirements. Default roles cannot be deleted.

Prerequisites

The current logon user has the system administrator role ADMIN or the role management role ROLE_MANAGER and the corresponding permissions.

Procedure

1. Log on to OCP.
2. In the left-side navigation pane, choose **System Management > Security**.
3. On the **Role Management** tab, find the specified role, and click **Delete** in the corresponding **Actions** column.



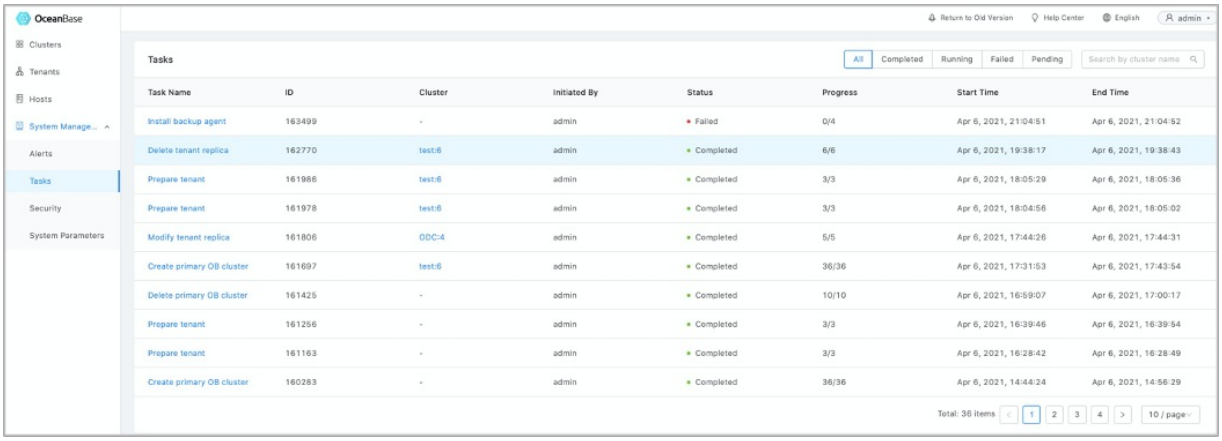
4. In the dialog box that appears, click **Delete**.

17.1.3.8. Manage tasks

17.1.3.8.1. View a task list

This topic describes the information that you can view in a task list.

By default, the **Tasks** page displays all the tasks in the clusters that the current logon user has permissions to view. The list displays the following task information: **Task Name**, **ID**, **Cluster**, **Initiated By**, **Status**, **Progress**, **Start Time**, and **End Time**. Among the information, **Cluster** is displayed in the format of Cluster name:Cluster ID. On the **Tasks** page, you can filter tasks by task status. The task statuses are **Completed**, **Running**, **Failed**, and **Pending**. By default, tasks in all statuses are displayed. You can also search for all the relevant tasks of a specified cluster by cluster name.



17.1.3.8.2. View task details

In the left-side navigation pane, choose System Management > Tasks. On the Tasks page, you can click the specified task name to view the task details.

On the task details page, a topology displays the upstream and downstream relationships of each step and the status of each step. You can move the topology by dragging and dropping the pointer in a blank area, and zoom out or zoom in the topology by scrolling the mouse wheel. You can also adjust the size of the canvas by using the zoom control in the upper-right corner. The topology is zoomed out or zoomed in by centering on the mouse pointer. For a long task flow, you can first zoom out the topology to quickly locate the desired step, and then zoom in the topology. This helps you quickly locate the step.

Retry or abort failed tasks

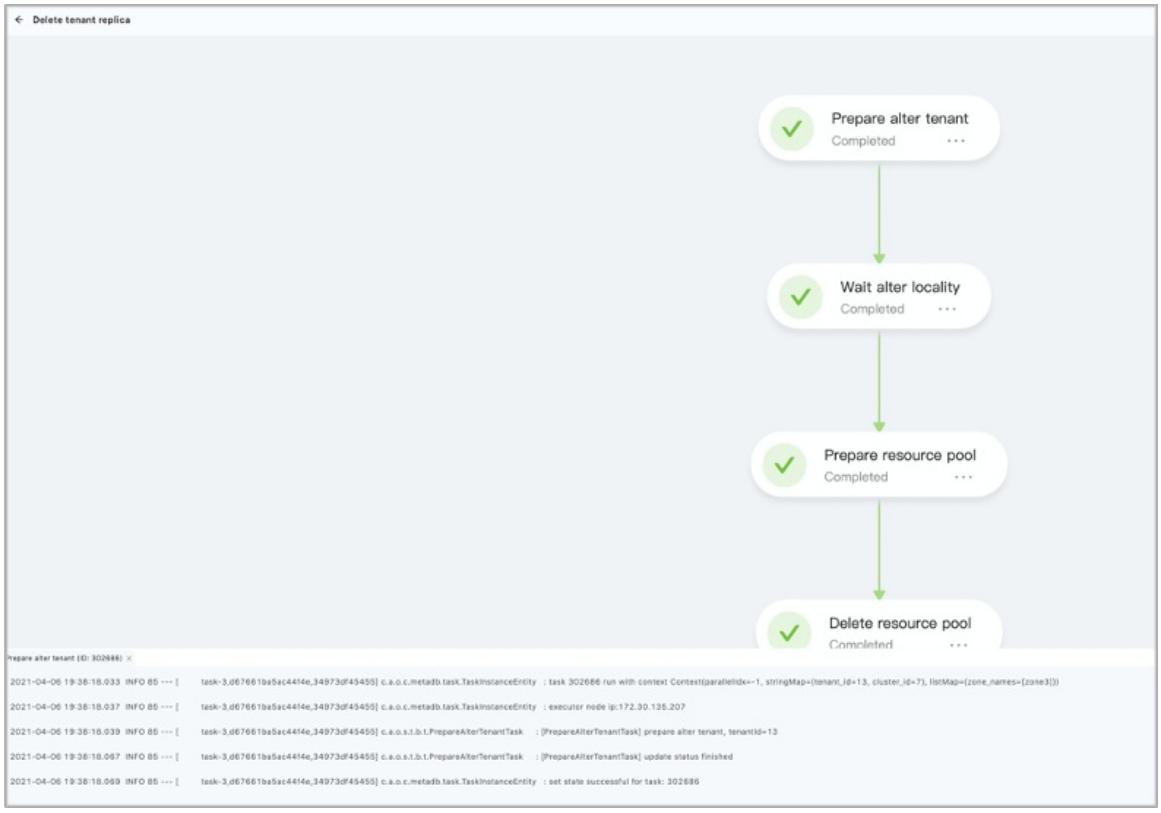
For failed tasks, OCP provides the Retry or Abort Task operation in the upper-right corner of a task topology.

- Abort Task: Perform a rollback operation for each step from the failed node to the root node of the task. This is primarily used to reclaim the resources and modify the status of the task.
- Retry: Perform the rollback and retry steps in sequence, starting from the failed node. If the operations are successful, continue to run the downstream nodes.

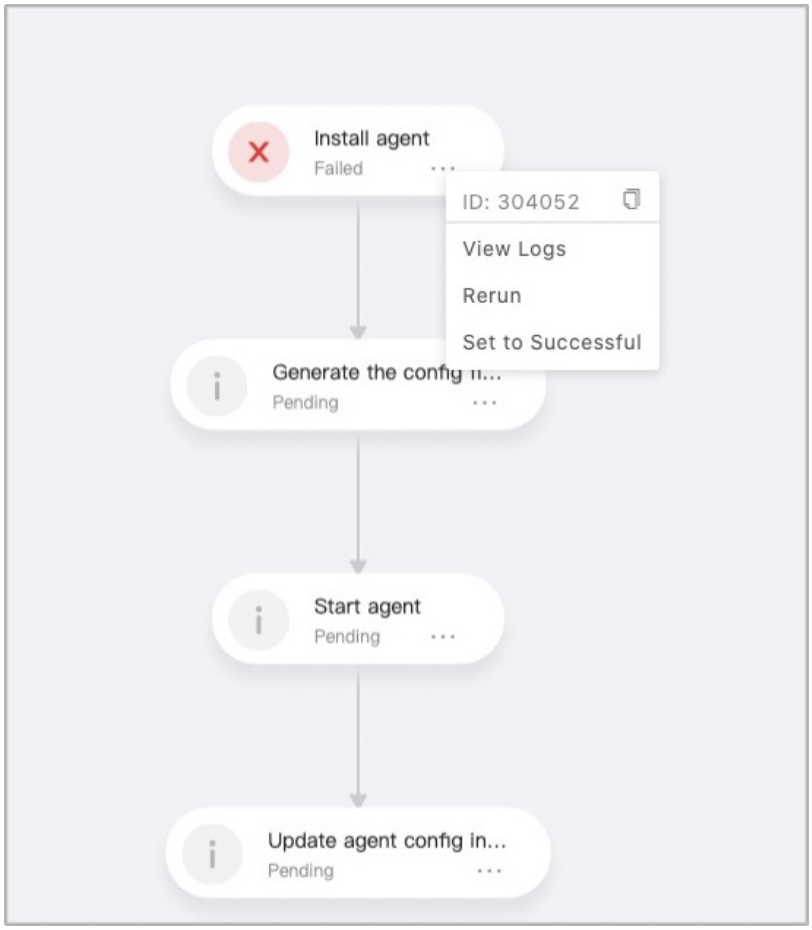
View the details of steps

You can perform different operations based on the node status for the specified step of a task. For example, you can view the step ID and logs. You can expand the hidden menu on the right of a specific step to view the operations that are supported by the current node.

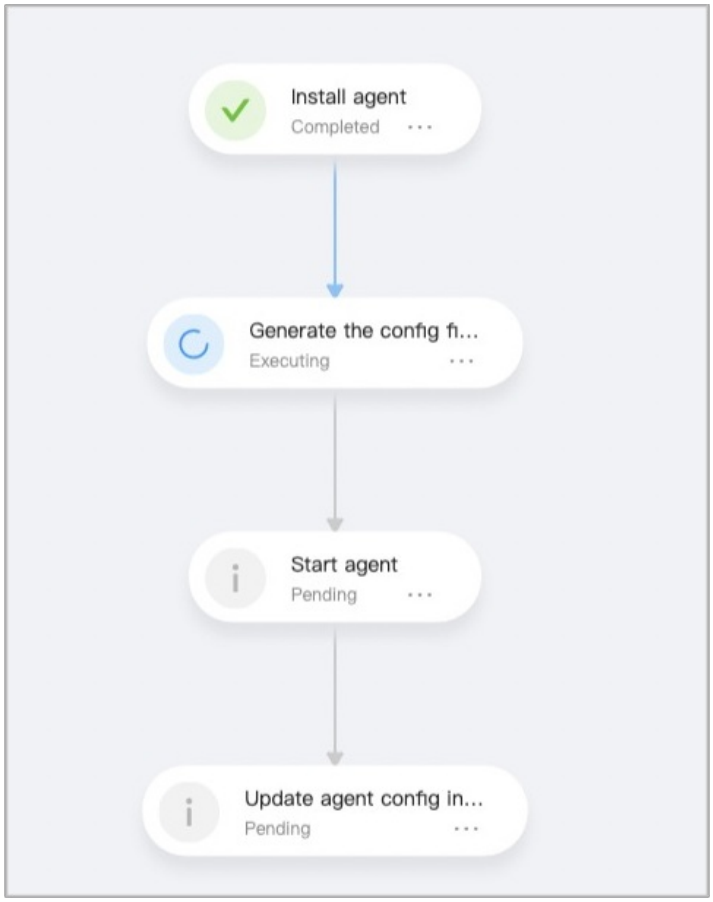
- For nodes whose status is successful, you can query logs. After you unfold the menu, the log information appears in the window at the bottom of the screen. You can enlarge the window by dragging the window.



- For nodes whose status is failed, you can perform the Rerun and Set to Successful operations in addition to viewing log information. Rerun indicates that you run the current node again. This means that you perform the rollback operation first and then rerun the current node. If the operations are successful, continue to run downstream nodes. Set to Successful indicates that you can set the status of the current node to successful and continue to run downstream nodes.



- For running nodes, you can view logs or stop running the nodes. After a node stops running, the status of the node becomes failed.



- For nodes that are pending for running, you cannot perform an operation.

17.1.3.9. Manage system parameters

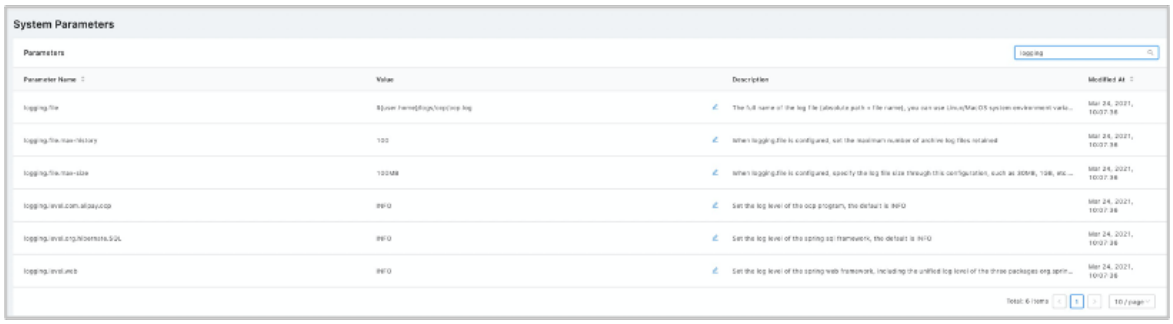
17.1.3.9.1. View system parameters

In OCP, system parameters are generally used to configure features, such as account logon and backup and recovery.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the left-side navigation pane, choose **System Management > System Parameters**. The System Parameters page displays a list of all the parameters for which OCP allows you to change values.
3. In the search box in the upper-right corner of the page, enter the name of the system parameter you need to view or a parameter keyword. Then, click the search icon to search for the specified system parameter.
   - The page displays the following parameter information: **Parameter Name**, **Value**, **Description**, and **Modified At**.
   - The parameter list also supports paging queries.

- The parameter description includes details, such as the parameter usage, valid values, and whether to restart to make the parameter take effect.

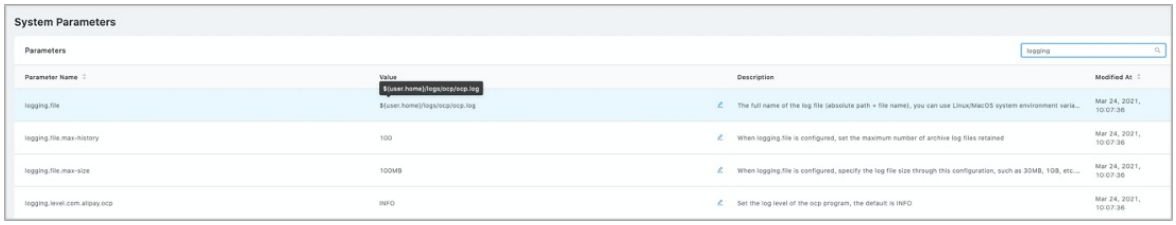


17.1.3.9.2. Modify system parameters

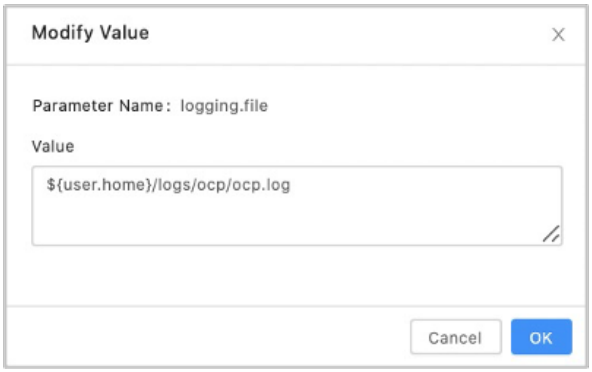
System parameters in OCP are generally used to configure features, such as account logon and backup and recovery. You can modify the relevant settings by changing the values of the system parameters.

Procedure

1. Log on to OCP. By default, the **Cluster Overview** page appears.
2. In the left-side navigation pane, choose **System Management > System Parameters**. The System Parameters page displays a list of all the parameters for which OCP allows you to change values.
3. In the search box in the upper-right corner of the page, enter the name of the system parameter you need to view or a parameter keyword. Then, click the search icon to search for the specified system parameter.
4. Click the Edit icon that corresponds to the specified parameter.



5. Enter a new parameter value.



6. Click **OK** to submit modifications.
   - The modifications take effect in about 10 minutes after the parameter is modified.
   - Some parameters require a restart to take effect. For more detailed descriptions about specific parameters, see Appendix 1. OCP configuration parameters.

17.1.3.10. Personal center

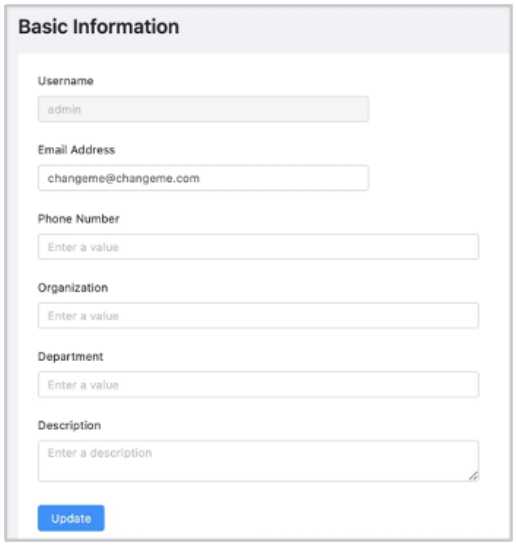
17.1.3.10.1. Specify personal information

After you log on to OCP, the current logon username appears in the upper-right corner of the system. You can click the current logon username to go to the personal center. In the personal center, you can specify your personal information, such as the contact information, the organization, and the department.

Procedure

1. In the upper-right corner of the OCP system interface, click the current logon username and select **Personal Settings**.

2. On the **Basic Information** page, specify your email address, phone number, organization, department, and personal description.



3. After you complete the settings, click **Update**.

17.1.3.10.2. Change the logon password

After you initially log on to OCP, you must change the password at the earliest opportunity to prevent an account leakage.

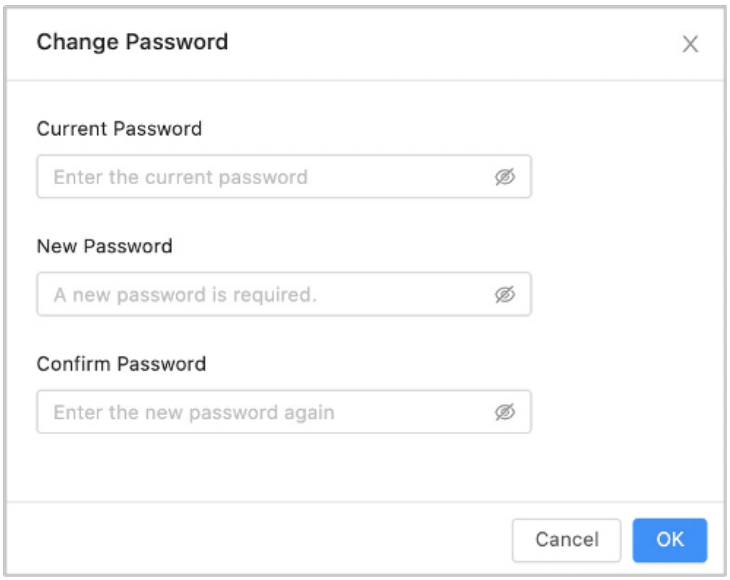
Background

We recommend that you regularly change your password to ensure account security.

Procedure

1. In the upper-right corner of the OCP system interface, click the current logon username and select **Change Password**.

2. In the **Change Password** dialog box, enter the old password and a new password and confirm the new password.



When you change the password, the new password must meet the requirements on the combination of uppercase letters, lowercase letters, and digits, and the length requirements.

- Must be 8 to 32 characters in length.
- Must contain at least two digits, two uppercase letters, two lowercase letters, and two special characters. Special characters are: _+@#$%

3. Click **OK**.

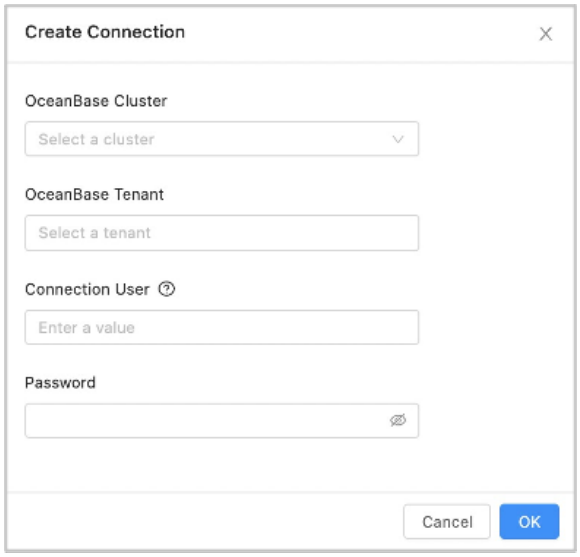
17.1.3.10.3. Password box

Personal password boxes are used to manage the connection information of the current logon user, such as the username and password for logging on to ApsaraDB for OceanBase, in a centralized manner. When you perform management operations on an ApsaraDB for OceanBase cluster or tenant in OCP, OCP must authenticate the connection information that corresponds to the password box of the current user.

Procedure

1. In the upper-right corner of the OCP system interface, click the current logon username and select **Password Box**. The **Password Box** page appears. For security purposes, the password on the page appears as ***.
2. Perform the following operations:

- To create a connection, click **Create Connection**. In the **Create Connection** dialog box, configure the relevant connection information and click **OK**.



- To delete connection information, select the connection information to be deleted, and then click **Delete** in the corresponding **Actions** column.
- Update connection information

> **Note**
>
> - To update the connection information of a cluster, you must delete the connection information first and add it again.
> - If you directly add connection information without deleting the original connection information, the new connection information does not overwrite the existing connection information. The system prompts that the duplicate record already exists.

17.1.3.11. Appendix

17.1.3.11.1. Appendix 1. OCP configuration parameters

This topic describes all the system parameters in OCP for the reference of OCP administrators. By default, the parameters automatically take effect in about 10 minutes after they are modified. Some parameters can take effect only after the OCP Server is started. For more information, see the special notes "Restart to take effect" included in the parameter description column.

Parameter	Default value	Parameter description

Parameter	Default value	Parameter description
ocp.active.standby.cluster.enabled	TRUE	Specifies whether to support the hot standby database feature of ApsaraDB for OceanBase. By default, this feature is enabled. By default, this feature is disabled when the old version of OCP is upgraded to version 2.4.x. If you need to enable this feature, you must make sure that the version of the OBProxy is not earlier than 1.7.2.
ocp.config-url.site.url		The Uniform Resource Locator (URL) for external access to the OCP config url service. By default, this parameter is empty. This indicates that the value that is the same as the value of ocp.site.url is used. If the config url service is separately deployed, you must specify a value that is different from the value of ocp.site.url.
ocp.env	private	The OCP environment. This configuration item is compatible with versions that are earlier than 2.3.x.
ocp.site.url	http://localhost:8080	The URL for external access to the OCP website. The URL must start with http instead of https and includes the Virtual IP Address (VIP), domain, and port. The URL cannot include a slash (/) at the end, such as http://localhost:8080.
ocp.system.default.language	zh-CN	The default language of the system (non-frontend language settings). If you do not specify this parameter, zh-CN is used. Restart to take effect.
ocp.system.default.timezone		The default time zone of the system. If you do not specify this parameter, the system default time zone is used. Restart to take effect.
ocp.version	2.4.2	The version number of OCP.
server.port	8080	Set the port number for the application server to start. Default value: 8080. Restart to take effect.

Parameter	Default value	Parameter description
springdoc.api-docs.enabled	FALSE	The OpenAPI 3 document API: specifies whether to enable the API. Default value: FALSE.

Security and logon module

Parameter	Default value	Parameter description
server.servlet.session.timeout	30m	The time-out period for logon sessions. Default value: 30m. The value of this parameter is at least 60s. If no suffix unit is added, the default unit is second.
ocp.iam.auth	local	The logon option of the web page. Valid values: local (OCP account in local MetaDB) and buc (employee account of Alibaba Group). Default value: local. Restart to take effect.
ocp.iam.auth.basic.enabled	TRUE	Specifies whether to enable the Basic Auth logon mode. Generally, this logon mode is used for clients, such as applications and SDKs. Default value: true. You can enable this configuration and ocp.iam.auth at the same time. Restart to take effect.
ocp.iam.auth.whitelist	/api/v2/time,.*Action=(ObRootServiceInfo|ObRootServiceRegister|ObIDCRegionInfo|GetObProxyConfig|AllocateClusterIdByNameAndIdx|DeleteObRootServiceInfoByClusterName).*	The whitelist of OCP logon-free APIs. The value is a comma-separated URL string that can be matched by regular expressions. Default value: /api/v2/time. Restart to take effect.
ocp.iam.csrf.enabled	TRUE	Specifies whether to enable cross-site request forgery (CSRF) security protection. In general, we recommend that you enable this feature for logons based on web pages. Default value: true. Restart to take effect.

Parameter	Default value	Parameter description
ocp.iam.csrf.url.excluded	/api/v2/iam/login.*,.*Action=(ObRootServiceInfo|ObRootServiceRegister|ObIDCRegionInfo|GetObProxyConfig|AllocateClusterIdByNameAndIdx|DeleteObRootServiceInfoByClusterName).*	You can specify a URL list to skip CSRF protection. Regular expression matching is supported. Default value: /api/v2/iam/login.*. This option only takes effect if CSRF security protection is enabled. The URL is a complete string that contains path and query param.
ocp.iam.login.lockout-minutes	30	Specifies the time when the logon from the client IP address is temporarily blocked if the logon failures limit is exceeded. Unit: min. The default value is 30 min.
ocp.iam.login.max-attempts	5	The number of consecutive username or password errors that are allowed during logon. If the limit is exceeded, the logon from the client IP address is temporarily blocked. The default value is five times.

OCP meta database connection (MetaDB)

Parameter	Default value	Parameter description
spring.datasource.druid.initialSize	10	The number of physical connections that are established during initialization. Restart to take effect.
spring.datasource.druid.keepAlive	TRUE	For connections within the number specified by minIdle in the connection pool, if the idle time exceeds minEvictableIdleTimeMillis (the default value is 1,800 seconds), the keepAlive operation is performed. Restart to take effect.
spring.datasource.druid.maxActive	100	The maximum number of connection pools. Restart to take effect.
spring.datasource.druid.maxWait	2000	The maximum waiting time when a connection is obtained. Unit: ms. Restart to take effect.

Parameter	Default value	Parameter description
spring.datasource.druid.minIdle	2	The minimum number of connection pools. Restart to take effect.
spring.datasource.druid.name	metadb-connect-pool	**MetaDB** druid connection pool name. Restart to take effect.
spring.datasource.druid.testWhileIdle	TRUE	Recommended configuration: **true** This does not affect performance and ensures security. It is checked when you apply for a connection. Restart to take effect.
spring.datasource.druid.validationQuery	SELECT 1 FROM DUAL	The SQL statement that is used to check whether the connection is valid. Restart to take effect.

OCP monitoring database connection (MonitorDB)

Parameter	Default value	Parameter description
ocp.monitordb.database	chacha_dev	The name of the database for storing monitoring data.
ocp.monitordb.driverClassName	com.alipay.oceanbase.obproxy.mysql.jdbc.Driver	The Java Database Connectivity (JDBC) driver. Restart to take effect.
ocp.monitordb.druid.initialSize	10	The number of physical connections that are established during initialization. Restart to take effect.
ocp.monitordb.druid.keepAlive	TRUE	For connections within the number specified by minIdle in the connection pool, if the idle time exceeds minEvictableIdleTimeMillis (the default value is 1,800 seconds), the keepAlive operation is performed. Restart to take effect.

Parameter	Default value	Parameter description
ocp.monitordb.druid.maxActive	100	The maximum number of connection pools. Restart to take effect.
ocp.monitordb.druid.maxWait	2000	The maximum waiting time when a connection is obtained. Unit: ms. Restart to take effect.
ocp.monitordb.druid.minIdle	2	The minimum number of connection pools. Restart to take effect.
ocp.monitordb.druid.name	monitordb-connect-pool	druid connection pool name. Restart to take effect.
ocp.monitordb.druid.testWhileIdle	TRUE	We recommend that you set the value to true. This does not affect performance and ensures security. It is checked when you apply for a connection. Restart to take effect.
ocp.monitordb.druid.validationQuery	SELECT 1 FROM DUAL	The SQL statement that is used to check whether the connection is valid. Restart to take effect.
ocp.monitordb.host	10.101.194.179	The hostname of the database for storing monitoring data.
ocp.monitordb.password	root	The password of the database for storing monitoring data.
ocp.monitordb.port	2888	The port of the database for storing monitoring data.
ocp.monitordb.username	root@ocp_meta#obocp	The username of the database for storing monitoring data.

Alert module

Parameter	Default value	Parameter description
ocp.alarm.send.batch-max	20	The maximum number of new notifications that are retrieved at a time.

Parameter	Default value	Parameter description
ocp.alarm.send.once-retry-times	3	The maximum number of retries when alarms are sent each time.
ocp.alarm.send.once-timeout-ms	10000	The time-out period for sending alarms each time.
ocp.alarm.send.period.ms	2000	The interval (ms) for sending notifications. The default value is 2,000 ms. Restart to take effect.
ocp.alarm.send.retry-timeout-minutes	60	The retry time-out period (min). If the period is exceeded, the system does not retry to send notifications. The default value is 60 min.
ocp.alarm.send.total-failed-retry-times	3	The maximum number of retries when notifications fail to be sent.
ocp.alarm.send.total-timeout-seconds	60	The time-out period for a single machine to send one notification.
ocp.backup.agent.relation.file.full-path.name	#{systemProperties['user.home'].concat('/ocp-server/etc/backup_agent_ob_relations_config.yaml')}	The file that describes the version of the backup and recovery component, the dependent metadatabase, and the mapping relationships between the version of the backup and recovery component and the version of ApsaraDB for OceanBase.

Backup and recovery module

Parameter	Default value	Parameter description
ocp.backup.alarm.backup-data-retention-days	7	The days that backup data is retained.
ocp.backup.alarm.backup-liboblog-expire-days	7	The latest days that the backup libblog is retained
ocp.backup.alarm.base-backup-last-finished-threshold	12960	The baseline fails to be backed up in the last nine days. The threshold is initially set to nine days (12,960 min). Unit: min

Parameter	Default value	Parameter description
ocp.backup.alarm.base-backup-timeout	10	The time-out period (min) for baseline backup scheduling.
ocp.backup.alarm.inc-backup-delay-threshold	3600	The alert threshold of the latency of incremental backup (in seconds).
ocp.backup.alarm.last-data-backup-max-interval-minutes	1440	Checks whether the baseline backup task fails in a recent time. The default value is one day. Unit: min.
backup.dbname.prefix	backup	The custom prefix of the metadatabase name of backup and recovery. The default prefix is backup. If the prefix is changed, only the names of new databases are affected during OCP initialization or upgrades, and the names of the existing databases are not changed.

Parameter	Default value	Parameter description
ocp.agent.rpc.port	62888	The service port of OCP-Agent. It is used by a remote procedure call (RPC).
ocp.ob-agent.version	t-oceanbase-ob-agent-2.4.0-1884988.alios7.x86_64.rpm	The OB-Agent version.
ocp.ocp-agent.version	t-oceanbase-ocp-agent-2.4.0-1884049.alios7.x86_64.rpm	The OCP-Agent version.

OCP-Agent and OB-Agent

Parameter	Default value	Parameter description
ocp.agent.rpc.port	62888	The service port of OCP-Agent. It is used by the RPC.
ocp.ob-agent.version	t-oceanbase-ob-agent-2.4.0-1884988.alios7.x86_64.rpm	The OB-Agent version.
ocp.ocp-agent.version	t-oceanbase-ocp-agent-2.4.0-1884049.alios7.x86_64.rpm	The OCP-Agent version.

OBProxy connections

Parameter	Default value	Parameter description
ocp.system.obproxy.address	localhost	The OBProxy address that is used to connect to the ApsaraDB for OceanBase cluster.
ocp.system.obproxy.metadb.cluster-name	obdv1	The cluster where OBProxy metadata resides. If MetaDB is not used when OBProxy is deployed, you can ignore this configuration.
ocp.system.obproxy.metadb.database	obproxy	The OBProxy metadatabase. If MetaDB is not used when OBProxy is deployed, you can ignore this configuration.
ocp.system.obproxy.metadb.user	root@obproxy	The user of the OBProxy metadatabase. If MetaDB is not used when OBProxy is deployed, you can ignore this configuration.
ocp.system.obproxy.port	2883	The OBProxy port that is used to connect to the ApsaraDB for OceanBase cluster.

Background tasks and scheduling

Parameter	Default value	Parameter description
ocp.task.executor.core-pool-size	16	The number of core threads in the task execution thread pool. Default value: 16. Restart to take effect.
ocp.task.executor.keep-alive-seconds	120	The time when idle threads in the task execution thread pool keep alive. Unit: s. The default value is 120s. Restart to take effect.
ocp.task.executor.max-pool-size	64	The maximum number of threads in the task execution thread pool. Default value: 64. Restart to take effect.

Parameter	Default value	Parameter description
ocp.task.executor.queue-capacity	1000	The size of the task execution thread queue. Default value: 1000. Restart to take effect.

Management of remote ApsaraDB for OceanBase connections

Parameter	Default value	Parameter description
obsdk.connectors.cache.cleanup.period.second	300	The interval for obsdk to clean up the connector cache. Unit: s. Valid values: 30 to 1800. Default value: 300.
obsdk.connectors.cache.max.idle.second	3600	The expiration time of the obsdk idle connector. Unit: s. Valid values: 300 to 18000. Default value: 3600.
obsdk.connectors.cache.size	50	The cache size of the obsdk connector. Valid values: 5 to 100. Default value: 50.
obsdk.connectors.print.sql	TRUE	Indicates whether to enable the SQL print in obsdk. By default, it is enabled.
obsdk.connectors.slow.query.threshold.millis	1000	The threshold of slow query logs in obsdk. Unit: ms. The default value is 1000 ms.
obsdk.ob.connection.mode	proxy	The connection mode of ApsaraDB for OceanBase. Valid values: proxy and direct. Default value: proxy.

Log module

Parameter	Default value	Parameter description
logging.file	${user.home}/logs/ocp/ocp.log	The full name (absolute path plus the file name) of the log file. You can use Linux or macOS system environment variables, such as ${HOME}, or the Java system variable ${user.home}. Default value: ${user.home}/logs/ocp/ocp.log.

Parameter	Default value	Parameter description
logging.file.max-history	100	After logging.file is configured, you can specify the maximum number of archived log files that are retained.
logging.file.max-size	100MB	After logging.file is configured, you can use this parameter to specify the log file size, such as 30 MB and 1 GB. Default value: 100MB.
logging.level.com.alipay.ocp	INFO	The log level of the OCP program. Default value: INFO.
logging.level.org.hibernate.SQL	INFO	The log level of the Spring SQL framework. Default value: INFO.
logging.level.web	INFO	The log level of the Spring Web framework. Default value: INFO.

Software package management

Parameter	Default value	Parameter description
ocp.file.default-block-split-size	1048576	The default file block size in the OCP file module. Unit: byte. The default value is 1 MB.
ocp.file.local.built-in.dir	#{systemProperties['user.home'].concat('/ocp-server/lib')}	The local path of the built-in files in the OCP file module.
ocp.file.local.dir	#{systemProperties['user.home'].concat('/data/files')}	The local path of the files in the OCP file module.
ocp.file.max-concurrent-count	16	The maximum number of files that are concurrently processed by a single node in the OCP file module. Default value: 16.
ocp.file.try-lock-timeout-milliseconds	10000	The lock time-out of a single node in the OCP file module, in milliseconds. Default value: 10000.

17.1.3.11.2. Appendix 2. Table of OCP resource unit specifications

This topic describes the built-in resource unit specifications in OCP. The minimum specification that is supported by ApsaraDB for OceanBase V2.0 or later is S1.

Specification name	Minimum number of CPUs	Maximum number of CPUs	Minimum memory	Maximum memory	Group
B	0.25	0.25	1 GB	1 GB	basic
LogOnlyNormal	1	1	2 GB	2 GB	logonly
LogOnlySystem	5	5	35 GB	35 GB	logonly
S0	0.5	0.5	2 GB	2 GB	standard
S1	1.5	1.5	6 GB	6 GB	standard
S2	3	3	12 GB	12 GB	standard
S3	6	6	20 GB	20 GB	standard
S4	12	12	40 GB	40 GB	standard

17.1.3.11.3. Appendix 3. Table of OCP error messages

This topic describes OCP error messages in a table.

Error code	Error message
1001	The passed parameter is empty or invalid: {0}. Check and try again.
1002	The record of the specified {0} type is not found. Parameter: {1}. Check and try again.
1003	
1004	Request processing timed out. Try again.

Error code	Error message
1005	A conflict occurs when the request is processed. Try again.
1006	The resource that has the same name already exists (type: {0}, parameter: {1}). Change to another name and try again.
1007	Unsupported unit: {0}
1010	The passed start and end time is invalid. Change the time and try again.
1011	Unsupported sorting field {0}.
1012	No configuration item found: {0}
1113	Failed to obtain the next value for the sequence.
1200	An exception occurs when RPC calls I/Os. server: {0} port: {1}
1201	Failed to call and execute RPC. server: {0} port: {1} command: {2} code: {3} result: {4}
1500	Failed to operate ApsaraDB for OceanBase. Error message: {0}
1998	
1999	An unknown error occurred. Cause: {0}, error message: {1}. Contact the administrator.
3000	The current user has not logged on or an error occurred during logon. Log on and try again.
3001	The current logon account has been disabled. Contact the administrator.
3002	The current logon account has expired. Contact the administrator.

Error code	Error message
3003	The current logon account has been locked. Contact the administrator.
3004	Invalid username or password. Try again. You have {0} remaining opportunities.
3005	The password has expired. Contact the administrator to reset the password.
3006	Invalid username or password. Try again. You have {0} remaining opportunities.
3007	Logon with the current account has been temporarily disabled. Contact the administrator.
3008	Invalid username {0}. The username must start with a lowercase letter and can contain lowercase letters, digits, underscores (_), and periods (.).The username must be 4 to 48 characters in length.
3009	Invalid password. The password must be 8 to 32 characters in length, and must contain at least two digits, two uppercase letters, two lowercase letters, and two special characters (_+@#$%)
3010	This operation requires the password of user {2} of tenant {1} in cluster {0}. Add the password to the password box first.
3011	The logon password of the account does not match the password provided. Check and try again.
3012	Invalid password ID format.
3013	The password whose ID is {0} does not exist in the password box.
3100	You have no permission to perform the corresponding operations. Contact the administrator.
3101	The expression of the permission cannot be empty. Check the permission settings and try again.

Error code	Error message
3102	The given expression {0} of the permission is invalid. Check the permission settings and try again.
3103	The system default role {0} cannot be deleted.
3104	The system default account {0} cannot be deleted.
3105	{0} is the username that is reserved by the system. Select another username and try again. Avoid including words, such as admin, sys, dba, and proxy.
4000	
4001	Task {0} does not exist.
4002	DAG {0} does not exist.
4003	A loop structure exists.
4004	Task context error.
4005	Failed to run the {0} command on host {1}.
4006	Failed to save file {0} on host {1}.
4007	Task {0} is in the {1} state. You cannot perform this operation.
5000	
6000	
7000	The specified resource {0}:{1} contains one or more child resources. Clear the child resources and try again.
7001	Host <{0}:{1}> already exists. Modify and try again.

Error code	Error message
7002	Failed to establish the SSH connection to the destination host {0}@{1}:{2} in the test. Cause: {3}. Modify the connection information and try again.
7004	The current host {0} is offline or in use. Release the host, recover the status of the host to {1}, and try again.
7005	Host <{0}:{1}> does not exist.
7006	This operation is not supported in the current host status {0}.
7007	The Internet data center (IDC) where host {0} resides does not match the IDC of the desired OBServer.
7008	The IDC where host {0} resides does not match the IDC of the desired zone.
7009	The region where host {0} resides does not match the region of the desired OBServer.
7010	Multiple {1} services of the same type cannot be deployed on host {0}. Select another host and try again.
7011	Multiple {1} services cannot be deployed on the host {0} of the container type. Select another host and try again.
7012	The agent service has been deployed on the current host {0}. Uninstall the agent and try again.
7013	The host whose IP address is <{0}> does not exist.
7014	The host whose IP address is <{0}> is not unique.
7500	File {0}:{1} does not exist.
7501	File block {0}:{1} is not found.
7502	The file that is named {1} and of the {0} type already exists.

Error code	Error message
7503	An error occurred while verifying the file suffix. The suffix is {0} but the expected suffix is {1}.
7504	An error occurred while verifying the real type of the file. The type is {0} but the expected type is {1}.
7505	Waiting for other requests to complete operating files timed out. Try again.
8000	Failed to access your personal center. Try again later.
8001	You are not allowed to delete the content of the password boxes of other users. Parameter: {0}
8002	The same connection information already exists. Delete the original record and create a connection again. Parameter: {0}
8500	The parameter {0} is empty or invalid:{1}. The root path of the website must start with http or https and contain the domain name and the port number. Check and try again.
10000	
11000	
11001	The specified cluster id={0} does not exist.
11002	The specified cluster name={0} does not exist.
11003	The specified cluster id={0}. The current type does not support this operation.
11004	The specified cluster id={0}. The current version does not support this operation. The minimum version that supports this operation is {1}.
11005	The information about the passed root server is invalid.

Error code	Error message
11007	This operation is not supported in the current status of cluster {0}.
11008	This operation is supported only when cluster {0} is in the {1} state.
11009	Invalid rootserver JSON {0}.
11010	No root server is found in the cluster whose ID is {0}.
11011	Invalid password. Make sure that the old password you enter is correct.
11012	The passed password parameter cannot be empty.
11013	The passed rpm package name cannot be empty.
11014	The cluster to be created already exists.
11015	The current cluster has a secondary cluster. This operation is not allowed.
11016	The secondary cluster does not support upgrade operations.
11017	No primary cluster is found in cluster {0}.
11018	Multiple primary cluster are found in cluster {0}
11019	The server list of cluster {0} is empty.
11020	The last O&M task of the cluster has not been completed.
11021	A cluster has been deployed on the passed machine.
11022	The hot secondary database feature of the cluster is not enabled.

Error code	Error message
11023	The specified cluster {0} does not meet the switching conditions.
11024	The addition operation is not completed in the specified secondary cluster {0}.
11030	The number of the passed machines does not match that of the OBServers to be imported to the cluster.
11031	The IP address of the passed machine does not match that of the OBServer to be imported to the cluster.
11032	Import the primary cluster first, and then import the secondary cluster.
11033	The cluster to be imported already exists.
11034	Region information about zone {0} is missing in the cluster to be imported.
11035	IDC information about zone {0} is missing in the cluster to be imported.
11050	The specified parameter {0} is not found.
11051	Format error for the value {1} of parameter {0}.
11052	When you specify a scope in which cluster parameters take effect, you can specify only a cluster, a zone, or a server. The current value {0} is invalid. Try again.
11100	Failed to connect to cluster {0}. Check whether the password of user {1} under the sys tenant of the cluster in the password box is correct. Check whether the whitelist of the sys tenant is correctly configured and whether the network is connected.
11201	The merged version number does not exist for the version {1} of the cluster whose ID is {0}.
11202	The merge query timed out. Try again.

Error code	Error message
11203	The name {0} of the metric for merging statistics is not supported.
11204	Failed to update the merge parameters because conflicting requests may exist. Check whether the update is successful a few seconds later.
11205	The merge operation cannot be initiated from a non-primary cluster. The cluster whose ID is {0} is a standby cluster.
11500	
11501	Failed to start ob agent on host {0}.
12000	
12001	The format of the entered zone name is invalid. The name must start with a letter, and can contain uppercase and lowercase letters, digits, and underscores (_). The name must be 2 to 64 characters in length.
12002	No zone {1} is found in the cluster whose ID is {0}.
12003	Zone {1} already exists in the cluster whose ID is {0}.
12004	This operation is not supported in the current status of the specified zone {0}.
12005	Failed to stop zone {0}.
12006	The specified zone {0} has a unit and cannot be deleted.
12007	The last O&M task in the zone has not been completed.
12008	The passed zone name already exists.
13000	
13001	The specified OBServer is not found. ID: {0}.

Error code	Error message
13002	The specified OBServer is not found. IP: {0}, port: {1}.
13003	This operation is not supported in the current status of the specified OBServer {0}.
13004	The observer {0} has not exited.
13005	Failed to start observer {0}.
13006	The IP address of the destination observer is invalid.
13007	The RPC port of the destination observer is invalid.
13008	The SQL port of the destination observer is invalid.
13009	The remaining number of servers cannot be less than the number of units of tenant {0} in zone {1}.
13010	The obagent {0} has not exited.
13011	The observer cannot be deleted. Cause: {0}.
13012	The last O&M task on the observer has not been completed.
13013	The specified OBServer is not found. IP: {0}.
13014	The specified observer is not unique. IP: {0}.
15000	
15001	The specified specification ID {0} does not exist.
15002	The specified specification name {0} does not exist.
15003	The format of the entered specification name is invalid. The name must start with a letter, and can contain uppercase and lowercase letters, digits, and underscores (_). The name must be 2 to 32 characters in length.

Error code	Error message
15004	The entered specification name {0} already exists.
15005	The entered CPU value is invalid. The minimum value is 0.5.
15006	The entered memory value is invalid. The minimum value is 1,073,741,824. Unit: byte.
15007	The entered disk value is invalid. The minimum value is 10,737,418,240. Unit: byte.
15008	The entered IOPS value is invalid. The minimum value is 128.
15009	
15010	The entered unit specification type is invalid. Valid values: SYSTEM and CUSTOM.
15020	The format of the entered tenant name is invalid. The name must start with a letter, and can contain uppercase and lowercase letters, digits, and underscores (_). The name must be 2 to 64 characters in length.
15021	Tenant {1} in cluster {0} already exists.
15022	Tenant {1} is not found in cluster {0}.
15023	The specified tenant ID {0} does not exist.
15024	The specified tenant ID {0} has been deleted.
15025	This operation is not supported in the current status of the specified tenant {0}.
15026	The specified tenant {0} is not allowed to be deleted.
15027	The operation is not allowed for the specified tenant {0}.
15028	Failed to set the password for the specified tenant whose ID is {0}.

Error code	Error message
15030	The entered tenant mode is invalid. Valid values: MYSQL and ORACLE.
15031	The entered length of the primary zone is invalid. The value must be less than 128.
15032	The entered primary zone is invalid.
15033	The entered whitelist length is invalid. The value must be less than 65,535.
15034	The entered password length is invalid. The value must be less than 32.
15035	The entered description length is invalid. The value must be less than 1,024.
15036	The minimum version that supports the Oracle mode is {0}. The current version is {1}.
15037	The passed zone is invalid
15040	The passed zone {0} is invalid.
15041	No zone {0} is found.
15042	No resource pool is available in the specified zone {0}.
15043	The specified unit {0} is not found.
15044	The specified unit {0} is the only resource unit in the zone to which the unit belongs and cannot be deleted.
15045	The passed number of units is invalid.
15046	The passed number of units cannot exceed the number of active servers in the zone.
15050	No server is available under tenant {0}.

Error code	Error message
15060	The specified tenant parameter {0} is not found.
15061	The specified tenant parameter {0} is of the read-only type.
15062	Failed to obtain the old value of the specified tenant parameter {0}.
15080	The passed replica type is invalid.
15082	The locality modification progress for tenant {0} is not found.
15083	Locality has not been modified for tenant {0}.
15084	The resource pool has not been scaled down for tenant {0}.
15100	Failed to connect to tenant {0}. Check whether the password of user root under the tenant in the password box is correct and whether the whitelist of the tenant is correctly configured.
20000	Backup and recovery is not supported in the {0} version of ApsaraDB for OceanBase.
20001	The configuration file {0} exists. However, no backup component applies to the cluster {2} of version {1}.
20002	The installation directory {1} whose IP address is {0} already exists. Check it manually.
20003	Backup scheduling needs to be configured first for cluster {0}.
20004	Parameter error.
20005	The backup configuration that apply to the cluster {1} of version {0} is not configured in the component. Configure it.

Error code	Error message
20006	None of the components {0} that apply to this cluster are online. Check it.
20007	The baseline backup task of the cluster or the tenant already exists.
20008	Immediate backup of the cluster failed.
20009	Failed to initiate data restoration.
20010	The URI in the backup configuration {0} is empty.
20011	The endpoint in the backup configuration {0} is empty.
20012	The accessKeyId in the backup configuration {0} is empty.
20013	The secretAccessKey in the backup configuration {0} is empty.
20014	The backup version number of tenant {0} is not found in the backup file.
20015	Failed to resolve the tenant information in the backup file.
20016	Failed to add the configuration file.
20017	Failed to delete the configuration because the configuration is being used by component {0}.
20018	The configuration name {0} already exists.
20020	The {0} component failed.
20021	The host already has a running O&M task (TaskId:{0}). Wait until the task is completed before you initiate a new O&M operation.
20030	The data backup is not started.
20031	The log backup is not started.

Error code	Error message
20032	The same cluster or tenants in the same cluster can have only one scheduling configuration. Cluster {0} and tenant {1} already exist. Delete the existing configuration before you add a configuration.
20040	Failed to create ResourceUnit {0} in the process of recovering the tenant. {1}
20041	The {0} format in the entered pool list is invalid. Check it.
20042	The pool in the entered pool list already exists. Change the pool name.
20043	The submitted new tenant {0} is being recovered. Check it.
20050	A switchover between primary and secondary databases may occur in cluster {0}. The secondary database {1} cannot be backed up. Back up the current primary database {1} to a new backup directory.
21000	
22000	

17.1.3.11.4. Appendix 4. OCP alert template variables

Alert template variables are used to configure alert rules and alert channels.

Category	Name	Description
Alert rule	app_type	Application type
Alert rule	alarm_type	Alert type
Alert rule	alarm_target	Alert object
Alert rule	alarm_scope	Scope
Alert rule	alarm_level	Alert severity

Category	Name	Description
Alert rule	alarm_evaluation_interval	Alert evaluation interval (second)
Alert rule	alarm_duration	Lasting duration (second)
Alert rule	alarm_status	Alert status
Alert rule	alarm_active_at	Time when the alert is triggered
Alert rule	alarm_resolved_at	Time when the alert is eliminated
Alert rule	alarm_last_interval	Lasting duration (second)
Alert rule	alarm_name	Alert name
Alert rule	value	Indicator value
Alert rule	alarm_threshold	Alert threshold
Alert rule	alarm_updated_at	Time when the alert is updated
Alert rule	service	Service name
Alert rule	ob_cluster_group	ApsaraDB for OceanBase cluster group
Alert rule	ob_cluster	ApsaraDB for OceanBase cluster
Alert rule	ob_cluster_id	ID of the ApsaraDB for OceanBase cluster
Alert rule	ob_tenant	ApsaraDB for OceanBase tenant
Alert rule	host_ip	IP address of the host
Alert rule	app_cluster	Application cluster
Notification message	app_type	Application type

Category	Name	Description
Notification message	alarm_type	Alert type
Notification message	alarm_target	Alert object
Notification message	alarm_scope	Scope
Notification message	alarm_level	Alert severity
Notification message	alarm_evaluation_interval	Alert evaluation interval (second)
Notification message	alarm_duration	Lasting duration (second)
Notification message	alarm_status	Alert status
Notification message	alarm_active_at	Time when the alert is triggered
Notification message	alarm_resolved_at	Time when the alert is eliminated
Notification message	alarm_last_interval	Lasting duration (second)
Notification message	alarm_name	Alert name
Notification message	value	Indicator value
Notification message	alarm_threshold	Alert threshold
Notification message	alarm_updated_at	Time when the alert is updated
Notification message	alarm_summary	Alert overview
Notification message	alarm_description	Alert details
Notification message	alarm_level_color	Color that corresponds to the alert severity
Notification message	alarm_id	Alert ID

Category	Name	Description
Notification message	alarm_url	URL of alert access on the OCP site
Notification message	service	Service name
Notification message	ob_cluster_group	ApsaraDB for OceanBase cluster group
Notification message	ob_cluster	ApsaraDB for OceanBase cluster
Notification message	ob_cluster_id	ID of the ApsaraDB for OceanBase cluster
Notification message	ob_tenant	ApsaraDB for OceanBase tenant
Notification message	host_ip	IP address of the host
Notification message	app_cluster	Application cluster
Aggregate message	app_type	Application type
Aggregate message	alarm_type	Alert type
Aggregate message	alarm_target	Alert object
Aggregate message	alarm_scope	Scope
Aggregate message	alarm_level	Alert severity
Aggregate message	alarm_evaluation_interval	Alert evaluation interval (second)
Aggregate message	alarm_duration	Lasting duration (second)
Aggregate message	alarm_status	Alert status
Aggregate message	alarm_active_at	Time when the alert is triggered
Aggregate message	alarm_resolved_at	Time when the alert is eliminated

Category	Name	Description
Aggregate message	alarm_last_interval	Lasting duration (second)
Aggregate message	alarm_name	Alert name
Aggregate message	value	Indicator value
Aggregate message	alarm_threshold	Alert threshold
Aggregate message	alarm_updated_at	Time when the alert is updated
Aggregate message	alarm_summary	Alert overview
Aggregate message	alarm_description	Alert details
Aggregate message	alarm_level_color	Color that corresponds to the alert severity
Aggregate message	alarm_id	Alert ID
Aggregate message	service	Service name
Aggregate message	ob_cluster_group	ApsaraDB for OceanBase cluster group
Aggregate message	ob_cluster	ApsaraDB for OceanBase cluster
Aggregate message	ob_tenant	ApsaraDB for OceanBase tenant
Aggregate message	host_ip	IP address of the host
Aggregate message	app_cluster	Application cluster
Aggregate message	alarm_group_by	Alert aggregation and grouping
Aggregate message	alarm_count	Number of Alerts
Channel configuration	app_type	Application type

Category	Name	Description
Channel configuration	alarm_type	Alert type
Channel configuration	alarm_target	Alert object
Channel configuration	alarm_scope	Scope
Channel configuration	alarm_level	Alert severity
Channel configuration	alarm_name	Alert name
Channel configuration	alarm_summary	Alert overview
Channel configuration	alarm_description	Alert details
Channel configuration	service	Service name
Channel configuration	ob_cluster_group	ApsaraDB for OceanBase cluster group
Channel configuration	ob_cluster	ApsaraDB for OceanBase cluster
Channel configuration	ob_tenant	ApsaraDB for OceanBase tenant
Channel configuration	host_ip	IP address of the host
Channel configuration	app_cluster	Application cluster
Channel configuration	message	Message (generated by the message template or the aggregate message template)
Channel configuration	message_json	JSON string of the message (the JSON format of the value of the message)
Channel configuration	recipients_uids	List of recipient user IDs that are separated with commas (,)
Channel configuration	at_recipients	The value of the @ recipient list

Category	Name	Description
Channel configuration	recipients_json_array	JSON array of the recipient list. It is used to @ recipients in DingTalk group messages

17.1.3.11.5. Appendix 5. List of background tasks

This topic describes the internal resident background tasks in OCP. These background tasks are generic terms for a category of tasks that are regularly triggered and executed by the OCP Server. The background tasks are primarily used for the purposes, such as maintaining OCP, checking the status of the managed objects, and cleansing data.

For information about the scheduling and running history of background tasks, see the details of the dag_instance and task_instance tables in the OCP MetaDB. You can perform a join query by using dag_instance.id = task_instance.dag_id). The dag_instance table records the execution details of the entire background tasks, whereas the task_instance table records the execution details of each subtask under the current background task.

You can search for the execution logs of background tasks by specific keyword in the OCP log file that is specified by the logging.file configuration parameter. The default OCP log file is ${user.home}/logs/ocp/ocp.log.

The following list of resident background tasks includes information, such as the task name, task description, remarks, and status troubleshooting. The information can be used as a reference for OCP system administrators and a troubleshooting guide.

Task name	Task description	Scheduling expression	Troubleshooting - database	Troubleshooting - OCP log
Archive alarm history data	Archive expired alert masking configurations and notifications.	0 0 3 * * *	ocp2_alarm_filter (alert masking configuration) ocp2_alarm_notification (alert notification) ocp2_alarm_filter_history (masking configuration history) ocp2_alarm_notification_history (notification history)	"Archive expired alarm history data: result=archive [filter|notification] success, affectRows=xxx"

Task name	Task description	Scheduling expression	Troubleshooting - database	Troubleshooting - OCP log
Sync all cluster info	Regularly synchronize ApsaraDB for OceanBase cluster information that is managed by OCP and update the information to the ob_cluster table. The information can be updated only when operateStatus of the ob_cluster table is NORMAL and clusterStatus is RUNNING.	0 * * * * *	ob_cluster (cluster information that is managed by OCP)	"cluster sync success, cluster=[id=xx,name=xx,..]" -- Success "cluster sync failed, cluster=xxx, exceptionType=xxx, message=xxx" -- Exception -- Synchronize process logs. " syncForCluster start, clusterId=xx init clusterSyncContext, context=xx create clusterOperator, clusterId=xx get current from ob cluster, clusterId=xx updateZones done, clusterId=xx syncForCluster done, clusterId=xx "
Refresh config properties	Refresh the configuration information of OCP.	0 */10 * * * *	config_properties (configuration of OCP system parameters)	"Refresh config properties on [host:port] at [date & time] "

Task name	Task description	Scheduling expression	Troubleshooting - database	Troubleshooting - OCP log
Maintain partition for monitordb	Process the partitions of the monitor tenant. New partitions are created for the partition table based on specific policies and expired partitions are deleted every day.	0 0 1 * * *	Monitor DB database: ob_cluster_system_event (collected Root Service events of ApsaraDB for OceanBase) ob_metric_data_1 (monitoring data collected per second) ob_metric_data_60 (monitoring data collected every 60 seconds)	-- Succeeded in maintaining the partition. "partition maintain start partition maintain success" -- Failed to maintain the partition. "partition maintain failed [exception msg]"
Sync system event	Synchronize system events. Regularly synchronize Root Service events from the ApsaraDB for OceanBase cluster to ob_cluster_system_event of monitordb.	0 */10 * * * *	Monitor DB database: ob_cluster_system_event (collected Root Service events of ApsaraDB for OceanBase) Internal table of ApsaraDB for OceanBase: oceanbase.__all_rootservice_event_history	-- Synchronization success "Batch saved [Number of records] events to database for cluster [cluster id] Sync-ed [Number of records] system events for cluster [cluster id], startTime [Start time], endTime [End time], limit [limit], offset [offset] Finished sync system events: cluster [cluster id], startTime [Start time], endTime [End time] " -- Synchronization failure "An exception occurred when saving events [exception log]"

Task name	Task description	Scheduling expression	Troubleshooting - database	Troubleshooting - OCP log
Check all cluster status	Regularly check the alive status of the ApsaraDB for OceanBase cluster in the ob_cluster table (check by establishing a database connection in a test). If the check fails, error logs are generated and an alert is sent.	0 * * * * *	ob_cluster (cluster information that is managed by OCP)	-- The cluster can be connected. "check cluster status okay, clusterId=[cluster id], obVersion=[ob version]" -- The cluster cannot be connected. "check cluster status unavailable, clusterId=[cluster id], failedReason=[exception msg]" -- Alert type and keyword "ob_cluster_status_check_failed" + "cluster connect check"

Task name	Task description	Scheduling expression	Troubleshooting - database	Troubleshooting - OCP log
Check host status	Check whether pos_proxy on the host can execute the remote command (whoami). If the execution fails, an alert is sent, and the host status (OFFLINE) is modified. If the execution is successful, the status is modified to the normal status (AVAILABLE or ONLINE). Check the number of node_exporters on the host. If the number is inexact, exception messages are generated and an alert is sent.	0 * * * * *	compute_host.status compute_host_agent.last_available_time ocp_exporter_address	"Host agent check result: current OS user is root" -- Agent check success "Check exporter status: host [id] desired exporter count: [number], actual exporter count: [number]" -- Exporter check success "Host agent check failed: host [host id] reason [exception msg]" -- Agent check failure "Alarm: host_unavailable for host ip [ip address]" -- Exporter check failure "Alarm: no_enough_exporter for host [id]" -- Exporter check failure "host_unavailable" -- Alert type of the agent failure "no_enough_exporter" -- Alert type of the exporter failure

Task name	Task description	Scheduling expression	Troubleshooting - database	Troubleshooting - OCP log
Submit scheduled dag	Implement a feature similar to crontab on the host. The feature is used to submit user-defined scheduled tasks. The user-defined scheduling expression is checked and compared with the current actual situation. If the condition is met, the user-defined task is submitted. Otherwise, the corresponding record status in the dag_schedule table is updated to ABNORMAL.	0 * * * * *	dag_schedule (details about task scheduling) dag_instance (running details of background tasks)	-- No user-defined task exists. "no cron job available in dag_schedule" -- An error occurred while submitting the user-defined task. "failed to get template for class [java class name], got exception: [exception msg]"

Task name	Task description	Scheduling expression	Troubleshooting - database	Troubleshooting - OCP log
Clean ocp task log files	Clean task logs on the OCP host.	0 0 1 * * *		The Python script that is used by OCP to clean logs: 'user.home' -> '/ocp-server/python-task/src/task/ob_task_clean_log.py' The directory of task logs to be cleaned: 'user.home' -> '/logs/task/' The log file of the Python process that cleans logs: 'user.home' -> '/logs/task/task.2.xxx.log' The logs of OCP cleaning task exceptions: "failed to clean task log files: [exception msg]"
Collect all cluster compaction info	Collect compaction information about all clusters.	0 * * * * *	ob_cluster_compaction (compaction records of ApsaraDB for OceanBase clusters) ob_zone_compaction (compaction records of ApsaraDB for OceanBase zones) ob_server_compaction (compaction records of ApsaraDB for OceanBase servers) ob_tenant_compaction_stats (compaction records of ApsaraDB for OceanBase tenants)	-- Collection success "Update cluster [cluster id] compaction status: from [xx] to [xx]" "Compaction post collect done, clusterId= [cluster id], version= [version]" -- Collection failure "Skip collect for not RUNNING cluster, clusterId=[cluster id], status=[xx]" "Skip collect for version 1, the first compaction will be version 2" "Validate compaction version failed: [reason]"

Task name	Task description	Scheduling expression	Troubleshooting - database	Troubleshooting - OCP log
Sync tenant information	Synchronize tenant information.	30 0/2 * * * ?	ob_tenant (tenant information table of ApsaraDB for OceanBase)	-- Synchronization success "[SyncAllTenantInfoTask] begin to sync tenant." "[SyncAllTenantInfoTask] sync tenants of specified cluster, clusterId=[cluster id]" "[SyncAllTenantInfoTask] sync tenant finished." -- Synchronization failure "Zero tenants found for cluster [cluster id]"
Detect alarms on schedule	Alarm detector related to backup and recovery.	0 */1 * * * *	dag_schedule (details about task scheduling) dag_instance (running details of background tasks)	-- Execution success generate tasks, templateName=Detect alarms, nodeCount=1, taskCount=1 set state running for task: [task id] set state successful for task: [task id] -- Execution failure Generate exception call stacks. Keyword: BackupAlarmService

Task name	Task description	Scheduling expression	Troubleshooting - database	Troubleshooting - OCP log
Clean expired data and log files on schedule	The scheduled cleaner related to backup and recovery cleans expired data and logs.	0 0 0 * * *	dag_schedule (details about task scheduling) dag_instance (running details of background tasks)	-- Execution success generate tasks, templateName=Clean expired data and log files, nodeCount=1, taskCount=1 set state running for task: [task id] set state successful for task: [task id] -- Execution failure Generate exception call stacks. Keyword :BackupCleanService
Check system obproxy	Check the health status of the OBProxy of the system that is used by OCP.	0 * * * * *	OCP system parameters: ocp.system.obproxy.address ocp.system.obproxy.port	-- The OBProxy status of the system is as expected. Generate the log "system obproxy is running". -- The OBProxy status of the system is abnormal. Generate the log "system obproxy is down". At the same time, an alert is pushed. -- Failed to use the check method. Generate the exception call stack: “check connection of system obproxy failed. exception: xxx"

Scheduling expression

The CRON expression in the following format is used for scheduling. The expression consists of the following six fields:

```
second, minute, hour, day of month, month, day(s) of week
```

Fields and values of the CRON expression

Field name	Valid value	Supported special characters
second	0-59	,-*/
minute	0-59	,-*/
hour	0-23	,-*/
day of month	1-31	,-*/?
month	1-12 or JAN-DEC	,-*/
day of week	1-7 or MON-SUN	,-*/?

Examples of the expression:

Expression	Description
*0 0 * * * * *	0 seconds, 0 minutes: Run once every hour and start at 0 seconds, 0 minutes.
*/10 * * * * *	Run once every 10 seconds.
*0 0 8-10 * * * *	Run once at 8:00, 9:00, and 10:00 every day and start at 0 seconds, 0 minutes.
*0 0/30 8-10 * * * *	Run at 8:00, 8:30, 9:00, 9:30,10:00, and 10:30 every day.
*0 0 9-17 * * MON-FRI*	Run from 9:00 to 17:00 from Monday to Friday.
*0 0 9-17 * * 1-5*	Run from 9:00 to 17:00 from Monday to Friday. A digit is used to represent the day of the week. The digit 1 represents Monday and the digit 7 represents Sunday. This is slightly different from the expression of crontab.
0 0 0 25 12 ?	00:00:00 on December 25 every year

17.1.3.11.6. Appendix 6. Metrics

This topic describes metrics.

Metric group	Metric name	Metric description	Calculation expression
CPU usage	cpu_percent	The CPU usage	100 * (1 - sum(rate(node_cpu_seconds_total{mode="idle", @LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(node_cpu_seconds_total{@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS))
I/O throughput rate	read	The amount of data read each time	avg(rate(node_disk_read_bytes_total{@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / 1048576
I/O throughput rate	write	The amount of data written each time	avg(rate(node_disk_written_bytes_total{@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / 1048576
I/O time consumption	read	The average time that is consumed by reading data per second	1000000 * avg(rate(node_disk_read_time_seconds_total{@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
I/O time consumption	write	The average time that is consumed by writing data per second	1000000 * avg(rate(node_disk_write_time_seconds_total{@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
IOPS	read	The number of reads per second	avg(rate(node_disk_reads_completed_total{@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)

Metric group	Metric name	Metric description	Calculation expression
IOPS	write	The number of writes per second	avg(rate(node_disk_writes_completed_total{@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Linux system load	load1	The average system load in the last 1 minute	avg(node_load1{@LABELS}) by (@GBLABELS)
Linux system load	load15	The average system load in the last 15 minutes	avg(node_load15{@LABELS}) by (@GBLABELS)
Linux system load	load5	The average system load in the last 5 minutes	avg(node_load5{@LABELS}) by (@GBLABELS)
MEMStore	active	The size of the active MemStore	sum(sysstat_value{metric_group="sysstat",stat_id="130000",@LABELS}) by (@GBLABELS) / 1048576
MEMStore	limit	The limit of the MemStore	sum(sysstat_value{metric_group="sysstat",stat_id="130004",@LABELS}) by (@GBLABELS) / 1048576
MEMStore	total	The total size of the MemStore	sum(sysstat_value{metric_group="sysstat",stat_id="130001",@LABELS}) by (@GBLABELS) / 1048576
MEMStore	trigger	The threshold that triggers the major freeze operation	sum(sysstat_value{metric_group="sysstat",stat_id="130002",@LABELS}) by (@GBLABELS) / 1048576

Metric group	Metric name	Metric description	Calculation expression
QPS	all	The number of SQL statements that are processed per second	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40002",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40004",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40006",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40008",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40000",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
QPS	delete	The number of DELETE statements that are processed per second	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40008",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
QPS	insert	The number of INSERT statements that are processed per second	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40002",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)

Metric group	Metric name	Metric description	Calculation expression
QPS	replace	The number of REPLACE statements that are processed per second	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40004",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
QPS	select	The number of SELECT statements that are processed per second	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40000",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
QPS	update	The number of UPDATE statements that are processed per second	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40006",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)

Metric group	Metric name	Metric description	Calculation expression
SQL response time	all	The average processing time of each SQL statement on the server	(sum(rate(sysstat_value{metric_group="sysstat",stat_id="40003",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40005",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40007",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40009",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40001",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)) /(sum(rate(sysstat_value{ metric_group="sysstat",stat_id="40002",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40004",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40006",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40008",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40000",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS))

Metric group	Metric name	Metric description	Calculation expression
Category of SQL execution plans	distributed	The number of distributed execution plans that are processed per second	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40012",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Category of SQL execution plans	local	The number of local SQL execution plans that are processed per second	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40010",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Category of SQL execution plans	remote	The number of remote execution plans that are processed per second	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40011",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
TPS	trans_count	The number of transactions that are processed per second	sum(rate(sysstat_value{metric_group="sysstat",stat_id="30005",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Event wait_Time	wait_time	The average time that is consumed by wait events	sum(rate(time_wait{metric_group="waitevent",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(total_waits{metric_group="waitevent",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)

Metric group	Metric name	Metric description	Calculation expression
Event wait_Count	wait_count	The number of wait events per second	sum(rate(total_waits{metric_group="waitevent",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Response time of transactions	trans_time	The average processing time of each transaction on the server	sum(rate(sysstat_value{metric_group="sysstat",stat_id="30006",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(sysstat_value{metric_group="sysstat",stat_id="30005",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Number of transaction logs	log_count	The number of logs of transactions that are committed per second	sum(rate(sysstat_value{metric_group="sysstat",stat_id="30002",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Time consumption of transaction logs	sync_time	The average time that is consumed by synchronizing transaction logs among networks each time	sum(rate(sysstat_value{metric_group="sysstat",stat_id="30000",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(sysstat_value{metric_group="sysstat",stat_id="30001",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)

Metric group	Metric name	Metric description	Calculation expression
Time consumption of transaction logs	write_disk	The average time that is consumed by writing transaction logs to disks each time	sum(rate(sysstat_value{metric_group="sysstat",stat_id="80041",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(sysstat_value{metric_group="sysstat",stat_id="80040",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Volume of transaction logs	log_size	The size of logs of transactions that are committed per second	sum(rate(sysstat_value{metric_group="sysstat",stat_id="80057",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Memory	buffers	The size of the buffer cache of the kernel	avg(node_memory_Buffers_bytes{@LABELS}) by (@GBLABELS) / 1073741824
Memory	free	The size of available physical memory	avg(node_memory_MemFree_bytes{@LABELS}) by (@GBLABELS) / 1073741824
Memory	used	The size of used physical memory	(avg(node_memory_MemTotal_bytes{@LABELS}) by (@GBLABELS) - avg(node_memory_MemFree_bytes{@LABELS}) by (@GBLABELS) - avg(node_memory_Cached_bytes{@LABELS}) by (@GBLABELS) - avg(node_memory_Buffers_bytes{@LABELS}) by (@GBLABELS)) / 1073741824

Metric group	Metric name	Metric description	Calculation expression
Response time	all	The average processing time of each SQL statement on the server	(sum(rate(sysstat_value{metric_group="sysstat",stat_id="40003",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40005",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40007",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40009",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40001",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)) /(sum(rate(sysstat_value{metric_group="sysstat",stat_id="40002",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40004",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40006",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40008",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40000",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS))

Metric group	Metric name	Metric description	Calculation expression
Response time	delete	The average processing time of each DELETE statement on the server	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40009",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(sysstat_value{metric_group="sysstat",stat_id="40008",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Response time	insert	The average processing time of each INSERT statement on the server	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40003",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(sysstat_value{metric_group="sysstat",stat_id="40002",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Response time	replace	The average processing time of each REPLACE statement on the server	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40005",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(sysstat_value{metric_group="sysstat",stat_id="40004",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Response time	select	The average processing time of each SELECT statement on the server	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40001",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(sysstat_value{metric_group="sysstat",stat_id="40000",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)

Metric group	Metric name	Metric description	Calculation expression
Response time	update	The average processing time of each UPDATE statement on the server	sum(rate(sysstat_value{metric_group="sysstat",stat_id="40007",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(sysstat_value{metric_group="sysstat",stat_id="40006",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Capacity_Number of partitions	partition_count	The number of partitions	sum(partition_count{metric_group="all_meta_table",@LABELS}) by (@GBLABELS)
Capacity_Number of tables	table_count	The number of tables	max(table_count{metric_group="all_table",@LABELS}) by (@GBLABELS)

Metric group	Metric name	Metric description	Calculation expression
Query response time	all	The average processing time of each SQL statement on the server	(sum(rate(sysstat_value{metric_group="sysstat",stat_id="40003",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40005",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40007",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40009",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40001",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)) /(sum(rate(sysstat_value{metric_group="sysstat",stat_id="40002",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40004",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40006",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40008",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="40000",@LABELS} [@INTERVAL]) by (@GBLABELS)) by (@GBLABELS))

Metric group	Metric name	Metric description	Calculation expression
Number of active sessions	active_session	The number of active sessions	sum(active_sessions{metric_group="all_virtual_processlist",@LABELS}) by (@GBLABELS)
Wait events	wait_count	The number of wait events per second	sum(rate(total_waits{metric_group="waitevent",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Time that is consumed by wait events	wait_time	The average time that is consumed by wait events	sum(rate(time_wait{metric_group="waitevent",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(total_waits{metric_group="waitevent",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)

Metric group	Metric name	Metric description	Calculation expression
Time that is consumed by waiting for locks	wait_time	The average wait time for write locks	sum(rate(sysstat_value{metric_group="sysstat",stat_id="60023",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / (sum(rate(sysstat_value{metric_group="sysstat",stat_id="60021",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) + sum(rate(sysstat_value{metric_group="sysstat",stat_id="60022",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS))
Cache hit ratio	block_cache	The block cache hit ratio	100 * 1 / (1 + sum(rate(sysstat_value{metric_group="sysstat",stat_id="50009",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(sysstat_value{metric_group="sysstat",stat_id="50008",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS))
Cache hit ratio	plan_cache	The cache hit ratio of execution plans	100 * sum(rate(hit_count{metric_group="plan_cache_stat",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(access_count{metric_group="plan_cache_stat",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)

Metric group	Metric name	Metric description	Calculation expression
Cache hit ratio	row_cache	The row cache hit ratio	100 * 1 / (1 + sum(rate(sysstat_value{metric_group="sysstat",stat_id="50001",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(sysstat_value{metric_group="sysstat",stat_id="50000",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS))
Cache size	block_cache	The block cache size	sum(cache_size{metric_group="all_virtual_kvcache_info",cache_name="user_block_cache",@LABELS}) by (@GBLABELS) / 1048576
Cache size	plan_cache	The cache size of execution plans	sum(mem_used{metric_group="plan_cache_stat",@LABELS}) by (@GBLABELS) / 1048576
Cache size	row_cache	The row cache size	sum(cache_size{metric_group="all_virtual_kvcache_info",cache_name="user_row_cache",@LABELS}) by (@GBLABELS) / 1048576
Network throughput rate	receive	The amount of data that is received per second	avg(rate(node_network_receive_bytes_total{@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / 1048576

Metric group	Metric name	Metric description	Calculation expression
Network throughput rate	send	The amount of data that is sent per second	avg(rate(node_network_transmit_bytes_total{@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / 1048576
Wait queue of requests	queue_count	The average number of times that SQL statements enter in the wait queue per second	sum(rate(sysstat_value{metric_group="sysstat",stat_id="20001",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Time that is consumed by the wait queue of requests	queue_time	The wait time of SQL statements in the wait queue	sum(rate(sysstat_value{metric_group="sysstat",stat_id="20002",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS) / sum(rate(sysstat_value{metric_group="sysstat",stat_id="20001",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Lock wait	fail	The number of failed write lock waits	sum(rate(sysstat_value{metric_group="sysstat",stat_id="60022",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)
Lock wait	success	The number of successful write lock waits	sum(rate(sysstat_value{metric_group="sysstat",stat_id="60021",@LABELS}[@INTERVAL]) by (@GBLABELS)) by (@GBLABELS)

17.1.4. SQL Reference (MySQL Mode)

17.1.4.1. Elements

17.1.4.1.1. Data types

ApsaraDB for OceanBase supports the following data types:

- Numeric types
- Date and time types
- String types
- Large object types

Numeric types

Numeric types can be divided into three categories:

- Integer types: `BOOL` or `BOOLEAN`, `TINYINT`, `SMALLINT`, `MEDIUMINT`, `INT` or `INTEGER`, and `BIGINT`.
- Fixed-point types: `DECIMAL` and `NUMERIC`.
- Floating-point types: `FLOAT` and `DOUBLE`.
- Bit-value type: `BIT`.

You can use the `UNSIGNED` keyword to declare all data types as unsigned data types. This operation changes the value ranges of the data types.

When you define a numeric data type, you can specify the precision and scale parameters. The meanings of the precision and scale parameters may vary based on the data type. For more information, see the description of each data type.

Integer types

An integer data type stores exact values of a fixed length. The value range depends on the data type length and whether the values are signed. The precision only indicates the minimum display width. For more information, see ZEROFILL keyword. The following table provides the details.

Type	Length (bytes)	Range (signed)	Range (unsigned)
`BOOL` / `BOOLEAN` / `TINYINT`	1	[-2 ^ 7 , 2 ^ 7 - 1]	[0, 2 ^ 8 - 1]
`SMALLINT`	2	[-2 ^ 15, 2 ^ 15 - 1]	[0, 2 ^ 16 - 1]
`MEDIUMINT`	3	[-2 ^ 23, 2 ^ 23 - 1]	[0, 2 ^ 24 - 1]
`INT` / `INTEGER`	4	[-2 ^ 31, 2 ^ 31 - 1]	[0, 2 ^ 32 - 1]

Type	Length (bytes)	Range (signed)	Range (unsigned)
`BIGINT`	8	[-2 ^ 63, 2 ^ 63 - 1]	[0, 2 ^ 64 - 1]

`BOOL` is equivalent to `BOOLEAN`, and the two types are equivalent to `TINYINT(1)`.

`INT` is equivalent to `INTEGER`.

Fixed-point types

A fixed-point type stores exact values of variable lengths. The value range and precision depend on the precision and scale parameters and whether the values are signed. The precision parameter specifies the maximum total number of valid digits in a decimal number. The scale parameter specifies the maximum number of digits after the decimal point in a decimal number. The maximum number of digits in the integer part is equal to the absolute difference between precision and scale. The maximum value of precision is 65. The maximum value of scale is 30. The default precision is 10. The default scale is 0. Examples

- `DECIMAL(5,2)` has three valid digits for the integer part and two valid digits for the fractional part. Therefore, the value range is [-999.99,999.99].
- If the value is `UNSIGNED`, the range is [0,999.99].

`DECIMAL` is equivalent to `NUMERIC`.

Floating-point types

A floating-point data type stores approximate numbers of a fixed length. The value range and precision depend on the type length, precision, scale, and whether the values are signed. The precision parameter specifies the maximum total number of valid digits in a decimal number. The scale parameter specifies the maximum number of valid digits after the decimal point in a decimal number. The maximum number of digits in the integer part is equal to the absolute difference between precision and scale. The maximum precision is 53. The maximum scale is 30.

The precision of a floating-point type is only a theoretical value based on the IEEE standard. The actual precision may vary due to hardware or operating system limits.

The following table lists the default settings when the precision and scale parameters are not specified.

Type	Length (bytes)	Range (signed)	Range (unsigned)	Precision
`FLOAT`	4	[-2 ^ 128, 2 ^ 128]	[0, 2 ^ 128]	7 digits
`DOUBLE`	8	[-2 ^ 1024, 2 ^ 1024]	[0, 2 ^ 1024]	15 digits

If the precision and scale parameters are specified, the value range is determined in the same way as the value range of a fixed-point type is determined.

ZEROFILL keyword

When you define a numeric type, you can use the ZEROFILL keyword to specify the minimum display width, and implicitly define the type as UNSIGNED. If the actual display width of the value is less than the minimum display width, the value is zero-padded to the minimum display width. The fractional part is padded with zeroes on the right to the width specified by the scale parameter. The integer part is padded with zeroes on the left to the width specified by the precision parameter. Examples

- `INT(5) ZEROFILL` : `123` is displayed as `00123` .
- `DECIMAL(10,5) ZEROFILL` : `123.456` is displayed as `00123.45600` .

Bit-value type

The BIT data type is used to store bit values. The BIT(M) data type can store M-bit values. M ranges from 1 to 64.

To specify bit values, use the b'value' notation. Set **value** to a binary number that consists of the digits 0 and 1. For example, b'111' indicates 7, and b'10000000' indicates 128.

When you insert a value into a BIT(M) column, if the length of the inserted value is less than **M**, the value is padded with zeros on the left. For example, the result of inserting b'101' into a BIT(6) column is equivalent to that of inserting b'000101'.

Date and time types

The following table provides the details.

Type	Syntax	Lower bound	Upper bound	Description
`DATETIME`	YYYY-MM-DD HH:MM:SS[.fraction]	0000-01-01 00:00:00.000000	9999-12-31 23:59:59.999999	The date and time. The time zone information is ignored.
`TIMESTAMP`	YYYY-MM-DD HH:MM:SS[.fraction]	0000-01-01 00:00:00.000000	9999-12-31 23:59:59.999999	The date and time. The time zone information is considered.
`DATE`	YYYY-MM-DD	0000-01-01	9999-12-31	The date.
`TIME`	HH:MM:SS[.fraction]	-838:59:59.000000	838:59:59.000000	The time.
`YEAR`	YYYY	1901	2155	The year.

The value ranges and precisions of `DATETIME` , `TIMESTAMP` , and `TIME` depend on the scale parameter. The scale parameter represents the maximum number of valid digits in the fractional part. The maximum value is 6, and the default value is 0.

String types

The following table describes common string types.

Type	Length	Maximum length in characters	Character set
VARCHAR	Variable length	262144 / 256K	UTF8MB4
VARBINARY	Variable length	1048576 / 1M	BINARY
CHAR	Fixed length	256	UTF8MB4
BINARY	Fixed length	256	BINARY
enum	Variable length	You can define a maximum of 65,535 elements and each element has a maximum length of 255 characters.	UTF8MB4
set	Variable length	You can define a maximum of 64 elements and each element has a maximum length of 255 characters.	UTF8MB4

Large object types

The following table provides information about large object types.

Type	Length	Maximum length in bytes	Character set
TINYTEXT	Variable length	256	UTF8MB4
TINYBLOB	Variable length	256	BINARY
TEXT	Variable length	65536 / 64K	UTF8MB4
BLOB	Variable length	65536 / 64K	BINARY

Type	Length	Maximum length in bytes	Character set
`MEDIUMTEXT`	Variable length	16777216 / 16M	`UTF8MB4`
`MEDIUMBLOB`	Variable length	16777216 / 16M	`BINARY`
`LONGTEXT`	Variable length	50331648 / 48M	`UTF8MB4`
`LONGBLOB`	Variable length	50331648 / 48M	`BINARY`

17.1.4.1.2. Expressions

Expressions are generalized notions. Each expression contains several input parameters and returns an output result. An input parameter may be a constant or a single row of data, or multiple rows of data. Expressions can be combined. The input of one expression can be the output of another expression.

Expressions can be divided into the following categories based on the source and form:

- Column reference
- Constant
- Operator
- Function

Examples

```
SELECT ABS(a + 1)
  FROM t1
 WHERE a > 0;
```

- a is a column reference.
- 0 and 1 are constants.
- `&gt;` and `+` are operators that take 0, 1, and the a expression as input.
- `ABS` is a function that takes the `+` expression as input.

17.1.4.1.3. Type conversion

ApsaraDB for OceanBase supports explicit type conversion and implicit type conversion.

You can perform explicit type conversion by using the `CAST` function.

Implicit type conversion occurs in the following scenario: An operation requires a parameter of a specified type, but the actual parameter value does not match the specified type. In this case, ApsaraDB for OceanBase converts the actual parameter value to the specified type before subsequent operations are performed.

17.1.4.1.4. Character sets

ApsaraDB for OceanBase supports the following character sets:

- `UTF8MB4`: variable-length encoding. The maximum length is 4 bytes.
- `BINARY`: fixed-length encoding. The fixed length is 1 byte.

`UTF8` (or `UTF8MB3`) is a subset of `UTF8MB4`. UTF8 is a variable-length encoding scheme that uses a maximum of 3 bytes to encode a character. To support seamless migration, ApsaraDB for OceanBase considers `UTF8` as the synonym of `UTF8MB4` in terms of the syntax.

17.1.4.1.5. Collations

ApsaraDB for OceanBase supports the following collations:

- `UTF8MB4_GENERAL_CI`
- `UTF8MB4_BIN`
- `BINARY`

17.1.4.1.6. Data comparison rules

ApsaraDB for OceanBase allows two or more types of data to be compared. The following list describes the possible comparison results:

- A non-zero value or TRUE
- 0 or False
- NULL

If the data types involved in a comparison are different, ApsaraDB for OceanBase automatically determines a data type for the comparison based on specific rules. Before comparison, the data is converted to this data type.

If the comparison type is text, you must also determine a collation for comparison.

17.1.4.1.7. Literals

Text literals

A text literal is a string that is enclosed by single quotation marks (`'`) or double quotation marks (`"`). If you enable the `ANSI_QUOTES` mode, you can use only single quotation marks (`'`) to enclose strings.

Numeric literals

Decimal numeric literals are divided into floating-point values and exact values that consist of integer and fixed-point values. Values can include a decimal point (`.`) as a decimal separator. Values that are prefixed with a minus sign (`-`) represent negative values.

Hexadecimal numeric literals only support integer values that are prefixed with `0X`, and allow letters from `A` to `F`. All letters are case-insensitive.

Datetime

Datetime literals can be in the text or numeric format.

- In the text format, you can use full separators, for example, `'2015-07-21 12:34:56.789'`. You can also use no separators, for example, `'20150721'`.
- The numeric format allows you to use only a decimal point (`.`) to separate the seconds and microseconds, for example, `20150721123456.789`.
- A decimal point (`.`) must be used between seconds and microseconds. If you require other separators, we recommend that you use only common separators such as hyphens (`-`), forward slashes (`/`), and colons (`:`).

Escape characters

An escape character is a backslash (`\`) that invokes an alternative interpretation on the following characters in a character sequence. Escape characters are case-sensitive. For example, `\b` represents the backspace and `\B` represents the `B` character.

The following table lists all escape characters.

Escape character	Description
`\b`	A backspace.
`\f`	A form feed.
`\n`	A line feed.
`\r`	A carriage return.
`\t`	A tab character.
`\\`	A backslash (\).
`\'`	A single quotation mark (').
`\"`	A double quotation mark (").

Escape character	Description
`\_`	An underscore (_).
`\%`	A percent sign (%).
`\0`	NULL
`\Z`	ASCII 26 (Ctrl+Z).

17.1.4.1.8. Comments

SQL statements

In general SQL statements, ApsaraDB for OceanBase supports the following three styles for comments:

- From a number sign (`#`) to the end of the line
- From a symbol (`--`) to the end of the line
- From a `/*` symbol to a `*/` symbol

Database objects

In a data definition language (DDL) statement, you can use the `COMMENT` clause to specify comments for a database object. Example:

```
create table t(pk INT PRIMARY KEY COMMENT 'Primary key');
```

17.1.4.2. Operators

17.1.4.2.1. Arithmetic operators

The following table lists all arithmetic operators.

Operator	Operand	Description
`+`	Unary or binary	As a unary operator, it indicates a positive number. As a binary operator, it indicates an addition.

Operator	Operand	Description
`-`	Unary or binary	As a unary operator, it indicates a negative number. As a binary operator, it indicates a subtraction.
`*`	Binary	The multiplication operator.
`/`	Binary	The division operator.
`DIV`	Binary	Divides an integer and returns the quotient.
`MOD` or `%`	Binary	Divides an integer and returns the remainder.

Integer division complies with the following rules:

- The quotient is rounded towards zero regardless of whether the quotient is positive or negative.
- The remainder has the same sign as the dividend.

Examples

```
OceanBase (root@oceanbase)> SELECT (-7) DIV (3.6), (-7) MOD (3.6);
+----------------+----------------+
| (-7) DIV (3.6) | (-7) MOD (3.6) |
+----------------+----------------+
|             -1 |           -3.4 |
+----------------+----------------+
1 row in set (0.01 sec)

OceanBase (root@oceanbase)> SELECT (-7) DIV (-3.4), (-7) % (-3.4);
+-----------------+---------------+
| (-7) DIV (-3.4) | (-7) % (-3.4) |
+-----------------+---------------+
|               2 |          -0.2 |
+-----------------+---------------+
1 row in set (0.02 sec)
```

17.1.4.2.2. Bitwise operators

`BIGINT UNSIGNED` is used for bitwise operations. The sign bits are ignored in the operations.

The following table lists all bitwise operators.

Operator	Operand	Description
&	Binary	Bitwise AND.
|	Binary	Bitwise OR.
~	Unary	Bitwise NOT.
^	Binary	Bitwise XOR.
<<	Binary	Logical left shift.
>>	Binary	Logical right shift.

17.1.4.2.3. Comparison operators

The following table lists all comparison operators.

Operator	Operand	Description	Effect of NULL on the operators
=	Binary	The equal operator.	The result is NULL.
<> / !=	Binary	The not equal operator.	The result is NULL.
>	Binary	The greater than operator.	The result is NULL.
>=	Binary	The greater than or equal operator.	The result is NULL.
<	Binary	The less than operator.	The result is NULL.
<=	Binary	The less than or equal operator.	The result is NULL.

Operator	Operand	Description	Effect of NULL on the operators
`[NOT] IN`	Binary	Specifies whether the value is in the set.	For more information, see the descriptions.
`[NOT] BETWEEN AND`	Ternary	Specifies whether the value is within the range.	For more information, see the descriptions.
`IS [NOT] TRUE`	Unary	Specifies whether the value is equal to TRUE.	The result is TRUE or FALSE.
`IS [NOT] FALSE`	Unary	Specifies whether the value is equal to FALSE.	The result is TRUE or FALSE.
`IS [NOT] NULL`	Unary	Specifies whether the value is equal to NULL.	The result is TRUE or FALSE.
`<=>`	Binary	The NULL-safe equal operator.	The result is TRUE or FALSE.

The following list describes the special treatment of the NULL value by some operators.

- `value [NOT] IN ()`:
  - If the value is NULL, the result is NULL.
  - If the value is not NULL and the set contains NULL, the result is TRUE when a non-NULL value in the set is equal to the value. Otherwise, the result is NULL.
- `value [NOT] BETWEEN lower AND upper`:
  - If the value is NULL or both the lower and upper values are NULL, the result is NULL.
  - Assume that the value is not NULL and the lower or upper value is NULL. In this case, the result that can be determined based on the value and the lower or upper value is returned.

Examples

```
OceanBase (root@oceanbase)> SELECT 1 IN (1, NULL), 1 IN (2, NULL);
+----------------+----------------+
| 1 IN (1, NULL) | 1 IN (2, NULL) |
+----------------+----------------+
|              1 |           NULL |
+----------------+----------------+
1 row in set (0.01 sec)

OceanBase (root@oceanbase)> SELECT 1 BETWEEN 0 AND NULL, 1 BETWEEN 2 AND NULL;
+----------------------+----------------------+
| 1 BETWEEN 0 AND NULL | 1 BETWEEN 2 AND NULL |
+----------------------+----------------------+
|                 NULL |                    0 |
+----------------------+----------------------+
1 row in set (0.01 sec)
```

17.1.4.2.4. Logical operators

The following table lists all logical operators.

Operator	Operand	Description
`AND` / `&amp;&amp;`	Binary	Logical AND
`OR` / `||`	Binary	Logical OR
`NOT` / `!`	Unary	Logical NOT

17.1.4.2.5. Date and time operators

The following table lists all date and time operators.

Operator	Operand	Description
`+`	Binary	The same as the `DATE_ADD` function.
`-`	Binary	The same as the `DATE_SUB` function.

Examples

```
OceanBase (root@oceanbase)> SELECT '2008-12-31 23:59:59' + INTERVAL 1 SECOND;
+-----------------------------------------+
| '2008-12-31 23:59:59' + INTERVAL 1 SECOND |
+-----------------------------------------+
| 2009-01-01 00:00:00                     |
+-----------------------------------------+
1 row in set (0.01 sec)
```

17.1.4.2.6. Concatenation operators

The following table lists all concatenation operators.

Operator	Operand	Description
`||`	Binary	Concatenates two strings.

Based on the preceding description, the `||` operator can also be used as a logical OR operator. The semantic meaning of the operator is controlled by the SQL mode.

- If the SQL mode is `PIPES_AS_CONCAT`, `||` is an operator that concatenates strings.
- If the SQL mode is not `PIPES_AS_CONCAT`, `||` is a logical OR operator.

17.1.4.2.7. Hierarchical query operators

The following table lists all hierarchical query operators.

Operator	Operand	Description
`PRIOR`	Unary	Indicates that the column is from the parent row.
`CONNECT_BY_ROOT`	Unary	Indicates the root ancestor.

17.1.4.2.8. Collation operators

The following table lists all collation operators.

Operator	Operand	Description
`COLLATE`	Unary (suffix)	The collation.

Examples

```
SELECT last_name
  FROM employees
 ORDER BY last_name COLLATE UTF8MB4_GENERAL_CI;
```

17.1.4.3. Functions

17.1.4.3.1. Functions

Date and time functions

Date and time functions are used to display information about dates and times.

CURDATE

Declaration

```
CURDATE()
```

Description

This function returns the current date without the exact time.

Example

```
OceanBase (root@test)> SELECT CURDATE();
+------------+
| CURDATE()  |
+------------+
| 2018-05-05 |
+------------+
1 row in set (0.00 sec)
```

CURRENT_DATE

Declaration

```
CURRENT_DATE()
CURRENT_DATE
```

Description

This function performs the same operation as `CURDATE()`.

CURRENT_TIME

Declaration

```
CURRENT_TIME([scale])
```

Description

This function returns the current time without the exact date.

`scale` specifies the precision of the fractional seconds. Valid values: 0 to 6. Default value: 0.

Example

```
OceanBase (root@test)> SELECT CURRENT_TIME(6);
+-----------------+
| CURRENT_TIME(6) |
+-----------------+
| 11:11:45.215311 |
+-----------------+
1 row in set (0.01 sec)
```

CURRENT_TIMESTAMP

Declaration

```
CURRENT_TIMESTAMP([scale])
```

Description

This function returns the current date and time based on the specified time zone.

`scale` specifies the precision of the fractional seconds. Valid values: 0 to 6. Default value: 0.

Example

```
OceanBase (root@test)> SELECT CURRENT_TIMESTAMP(6);
+----------------------------+
| CURRENT_TIMESTAMP(6)       |
+----------------------------+
| 2018-05-05 11:35:39.177764 |
+----------------------------+
1 row in set (0.01 sec)
```

CURTIME

Declaration

```
CURTIME()
```

Description

This function performs the same operation as `CURRENT_TIME()`.

DATE_ADD

Declaration

```
DATE_ADD(date, INTERVAL expr unit)
```

Description

This function performs arithmetic calculations on date and time values.

- `date` represents the date and time base. You must specify the date, whereas the time is optional.
- `expr` represents the time interval, which can be negative.
- `unit` represents the unit of the time interval.

The following table lists all units for time intervals.

Unit	Type	Description	Syntax
`MICROSECOND`	Independent	Microseconds	`MICROSECONDS`
`SECOND`	Independent	Seconds	`SECONDS`
`MINUTE`	Independent	Minutes	`MINUTES`
`HOUR`	Independent	Hours	`HOURS`
`DAY`	Independent	Days	`DAYS`
`WEEK`	Independent	Weeks	`WEEKS`
`MONTH`	Independent	Months	`MONTHS`
`QUARTER`	Independent	Quarters	`QUARTERS`
`YEAR`	Independent	Years	`YEARS`
`SECOND_MICROSECOND`	Combination	From seconds to microseconds	`'SECONDS.MICROSECONDS'`
`MINUTE_MICROSECOND`	Combination	From minutes to microseconds	`'MINUTES:SECONDS.MICROSECONDS'`
`MINUTE_SECOND`	Combination	From minutes to seconds	`'MINUTES:SECONDS'`
`HOUR_MICROSECOND`	Combination	From hours to microseconds	`'HOURS:MINUTES:SECONDS.MICROSECONDS'`
`HOUR_SECOND`	Combination	From hours to seconds	`'HOURS:MINUTES:SECONDS'`

Unit	Type	Description	Syntax
`HOUR_MINUTE`	Combination	From hours to minutes	`'HOURS:MINUTES'`
`DAY_SECOND`	Combination	From days to seconds	`'DAYS HOURS:MINUTES:SECONDS'`
`DAY_MINUTE`	Combination	From days to minutes	`'DDAYSD HOURS:MINUTES'`
`DAY_HOUR`	Combination	From days to hours	`'DAYS HOURS'`
`YEAR_MONTH`	Combination	From years to months	`'YEARS-MONTHS'`

Example

```
OceanBase (root@test)> SELECT
    -> DATE_ADD(NOW(), INTERVAL 5 DAY),
    -> DATE_ADD('2014-01-10', INTERVAL 5 MICROSECOND),
    -> DATE_ADD('2014-01-10', INTERVAL 5 SECOND),
    -> DATE_ADD('2014-01-10', INTERVAL 5 MINUTE),
    -> DATE_ADD('2014-01-10', INTERVAL 5 HOUR),
    -> DATE_ADD('2014-01-10', INTERVAL 5 DAY),
    -> DATE_ADD('2014-01-10', INTERVAL 5 WEEK),
    -> DATE_ADD('2014-01-10', INTERVAL 5 MONTH),
    -> DATE_ADD('2014-01-10', INTERVAL 5 QUARTER),
    -> DATE_ADD('2014-01-10', INTERVAL 5 YEAR),
    -> DATE_ADD('2014-01-10', INTERVAL '5.000005' SECOND_MICROSECOND),
    -> DATE_ADD('2014-01-10', INTERVAL '05:05.000005' MINUTE_MICROSECOND),
    -> DATE_ADD('2014-01-10', INTERVAL '05:05' MINUTE_SECOND),
    -> DATE_ADD('2014-01-10', INTERVAL '05:05:05.000005' HOUR_MICROSECOND),
    -> DATE_ADD('2014-01-10', INTERVAL '05:05:05' HOUR_SECOND),
    -> DATE_ADD('2014-01-10', INTERVAL '05:05' HOUR_MINUTE),
    -> DATE_ADD('2014-01-10', INTERVAL '01 05:05:05.000005' DAY_MICROSECOND),
    -> DATE_ADD('2014-01-10', INTERVAL '01 05:05:05' DAY_SECOND),
    -> DATE_ADD('2014-01-10', INTERVAL '01 05:05' DAY_MINUTE),
    -> DATE_ADD('2014-01-10', INTERVAL '01 05' DAY_HOUR),
    -> DATE_ADD('2014-01-10', INTERVAL '1-01' YEAR_MONTH)
    -> \G
*************************** 1. row ***************************
                                    DATE_ADD(NOW(), INTERVAL 5 DAY): 2018-05-10 14:54:52
                       DATE_ADD('2014-01-10', INTERVAL 5 MICROSECOND): 2014-01-10 00:00:00.000005
                            DATE_ADD('2014-01-10', INTERVAL 5 SECOND): 2014-01-10 00:00:05
                            DATE_ADD('2014-01-10', INTERVAL 5 MINUTE): 2014-01-10 00:05:00
                              DATE_ADD('2014-01-10', INTERVAL 5 HOUR): 2014-01-10 05:00:00
                               DATE_ADD('2014-01-10', INTERVAL 5 DAY): 2014-01-15
                              DATE_ADD('2014-01-10', INTERVAL 5 WEEK): 2014-02-14
                             DATE_ADD('2014-01-10', INTERVAL 5 MONTH): 2014-06-10
                           DATE_ADD('2014-01-10', INTERVAL 5 QUARTER): 2015-04-10
                              DATE_ADD('2014-01-10', INTERVAL 5 YEAR): 2019-01-10
       DATE_ADD('2014-01-10', INTERVAL '5.000005' SECOND_MICROSECOND): 2014-01-10 00:00:05.000005
   DATE_ADD('2014-01-10', INTERVAL '05:05.000005' MINUTE_MICROSECOND): 2014-01-10 00:05:05.000005
                DATE_ADD('2014-01-10', INTERVAL '05:05' MINUTE_SECOND): 2014-01-10 00:05:05
  DATE_ADD('2014-01-10', INTERVAL '05:05:05.000005' HOUR_MICROSECOND): 2014-01-10 05:05:05.000005
               DATE_ADD('2014-01-10', INTERVAL '05:05:05' HOUR_SECOND): 2014-01-10 05:05:05
                  DATE_ADD('2014-01-10', INTERVAL '05:05' HOUR_MINUTE): 2014-01-10 05:05:00
DATE_ADD('2014-01-10', INTERVAL '01 05:05:05.000005' DAY_MICROSECOND): 2014-01-11 05:05:05.000005
            DATE_ADD('2014-01-10', INTERVAL '01 05:05:05' DAY_SECOND): 2014-01-11 05:05:05
              DATE_ADD('2014-01-10', INTERVAL '01 05:05' DAY_MINUTE): 2014-01-11 05:05:00
                   DATE_ADD('2014-01-10', INTERVAL '01 05' DAY_HOUR): 2014-01-11 05:00:00
                  DATE_ADD('2014-01-10', INTERVAL '1-01' YEAR_MONTH): 2015-02-10
1 row in set (0.01 sec)
```

DATE_FORMAT

Declaration

```
DATE_FORMAT(date, format)
```

Description

This function returns the date and time in the specified format.

- `date` specifies the date and time.

- `format` specifies the output format.

The following table lists all output formats.

Format character	Description	Syntax
`%a`	The abbreviation of the day in a week.	`Sun..Sat`
`%b`	The abbreviation of the month.	`Jan..Dec`
`%c`	The numeric form of the month.	`1..12`
`%D`	The abbreviation of the day in a month.	`1st..31st`
`%d`	The numeric form of the day in a month.	`01..31`
`%e`	The numeric form of the day in a month.	`1.. 31`
`%f`	Microseconds.	`000000..999999`
`%H`	Hours.	`00..23`
`%h`	Hours.	`01..12`
`%I`	Hours.	`01..12`
`%i`	Minutes.	`00..59`
`%j`	The sequence number of the day in a year.	`001..366`
`%k`	Hours.	`0..23`

Format character	Description	Syntax
`%l`	Hours.	`0..12`
`%M`	The month name.	`January..December`
`%m`	The numeric form of the month.	`01..12`
`%p`	Morning or afternoon.	`AM/PM`
`%r`	Time in the 12-hour clock.	`hh:mm:ss AM/PM`
`%S`	Seconds.	`00..59`
`%s`	Seconds.	`00..59`
`%T`	Time in the 24-hour clock.	`hh:mm:ss`
`%U`	The sequence number of the week in a year when Sunday is the first day of each week.	`00..53`
`%u`	The sequence number of the week in a year when Monday is the first day of each week.	`00..53`
`%V`	The sequence number of the week in a year when Sunday is the first day of each week. This option is used in conjunction with `%X`.	`01..53`
`%v`	The sequence number of the week in a year when Monday is the first day of each week. This option is used in conjunction with `%x`.	`01..53`

Format character	Description	Syntax
`%W`	The name of the day in a week.	`Sunday..Saturday`
`%w`	The sequence number of the day in a week.	`0=Sunday..6=Saturday`
`%X`	The year of the week when Sunday is the first day of each week. This option is used in conjunction with `%V` .	
`%x`	The year of the week when Monday is the first day of each week. This option is used in conjunction with `%v` .	
`%Y`	The four-digit year.	
`%y`	The two-digit year.	
`%%`	The literal character `%` .	

Example

```
OceanBase (root@test)> SELECT
    -> DATE_FORMAT('2014-01-01', '%Y-%M-%d'),
    -> DATE_FORMAT('2014-01-01', '%X-%V'),
    -> DATE_FORMAT('2014-01-01', '%U')
    -> \G
*************************** 1. row ***************************
DATE_FORMAT('2014-01-01', '%Y-%M-%d'): 2014-January-01
   DATE_FORMAT('2014-01-01', '%X-%V'): 2013-52
      DATE_FORMAT('2014-01-01', '%U'): 00
1 row in set (0.01 sec)
```

DATE_SUB

Declaration

```
DATE_SUB(date, INTERVAL expr unit)
```

Description

This function performs arithmetic calculations on date and time values.

For more information, see `DATE_ADD()`.

DATEDIFF

Declaration

```
DATEDIFF(date1, date2)
```

Description

This function returns the number of days between `date1` and `date2`.

Only the date part of the parameter is used in the calculation, and the time part is ignored.

Example

```
OceanBase (root@test)> SELECT DATEDIFF('2015-06-19','1994-12-17');
+-------------------------------------+
| DATEDIFF('2015-06-19','1994-12-17') |
+-------------------------------------+
|                                7489 |
+-------------------------------------+
1 row in set (0.01 sec)
```

EXTRACT

Declaration

```
EXTRACT(unit FROM date)
```

Description

This function returns the specified portion of the `date` value as an integer value. If multiple portions are specified, all values are concatenated in order.

For information about the `unit` values, see `DATE_ADD()`. If the `unit` value is `WEEK`, see the description of `%U` in `DATE_FORMAT()`.

Example

```
SELECT EXTRACT(WEEK FROM '2013-01-01'),
EXTRACT(WEEK FROM '2013-01-06'),
EXTRACT(YEAR_MONTH FROM '2012-03-09'),
EXTRACT(DAY FROM NOW())\G;
* 1. row *
EXTRACT(WEEK FROM '2013-01-01'): 0
EXTRACT(WEEK FROM '2013-01-06'): 1
EXTRACT(YEAR_MONTH FROM '2012-03-09'): 201203
EXTRACT(DAY FROM NOW()): 18
1 row in set (0.00 sec)
```

FROM_DAYS

Declaration

```
FROM_DAYS(N)
```

Description

This function returns the `DATE` value based on the number of days `N`. N specifies the number of days that have elapsed since `0000-01-01`.

Example

```
OceanBase (root@test)> SELECT FROM_DAYS(736271), FROM_DAYS(700000);
+-------------------+-------------------+
| FROM_DAYS(736271) | FROM_DAYS(700000) |
+-------------------+-------------------+
| 2015-11-04        | 1916-07-15        |
+-------------------+-------------------+
1 row in set (0.00 sec)
```

FROM_UNIXTIME

Declaration

```
FROM_UNIXTIME(unix_timestamp)
FROM_UNIXTIME(unix_timestamp, format)
```

Description

- If you do not specify the `format` parameter, the function returns a `DATETIME` value that ignores the time zone.
- If you specify the `format` parameter, the function returns a date and time string in the specified format.

`unix_timestamp` specifies the UNIX timestamp. The value is the time that has elapsed since `1970-01-01 00:00:00.000000` in microseconds.

`format` supports the formats that are listed in the **description** of the `DATE_FORMAT()` function.

Example

```
OceanBase (root@test)> SELECT FROM_UNIXTIME(UNIX_TIMESTAMP(), '%Y %D %M %h:%i:%s %x');
+-----------------------------------------------------------+
| FROM_UNIXTIME(UNIX_TIMESTAMP(), '%Y %D %M %h:%i:%s %x') |
+-----------------------------------------------------------+
| 2018 5th May 08:41:26 2018                                |
+-----------------------------------------------------------+
1 row in set (0.01 sec)
```

MONTH

Declaration

```
MONTH(date)
```

Description

This function returns the month of the `date`.

Example

```
OceanBase (root@test)> SELECT MONTH('2008-02-03');
+---------------------+
| MONTH('2008-02-03') |
+---------------------+
|                   2 |
+---------------------+
1 row in set (0.01 sec)
```

NOW

Declaration

```
NOW([scale])
```

Description

This function performs the same operation as `CURRENT_TIMESTAMP()`.

PERIOD_DIFF

Declaration

```
PERIOD_DIFF(p1, p2)
```

Description

This function returns the interval between two dates in months. The date can contain only the year and month information, and the format must be `YYYYMM` or `YYMM`.

Example

```
OceanBase (root@test)> SELECT PERIOD_DIFF(200802, 200703);
+----------------------------+
| PERIOD_DIFF(200802,200703) |
+----------------------------+
|                         11 |
+----------------------------+
1 row in set (0.01 sec)
```

STR_TO_DATE

Declaration

```
STR_TO_DATE(str, format)
```

Description

This function converts a `str` value to a `DATETIME`, `DATE`, or `TIME` value based on the specified `format`. The data type of the returned value depends on the date and time information included in the specified `format`.

`format` supports the formats that are listed in the **description** of the `DATE_FORMAT()` function.

Example

```
OceanBase (root@test)> SELECT STR_TO_DATE('2014-Jan-1st 5:5:5 pm', '%Y-%b-%D %r');
+-----------------------------------------------------+
| STR_TO_DATE('2014-Jan-1st 5:5:5 pm', '%Y-%b-%D %r') |
+-----------------------------------------------------+
| 2014-01-01 05:05:05                                 |
+-----------------------------------------------------+
1 row in set (0.01 sec)
```

TIME

Declaration

```
TIME(datetime
```

Description

This function converts a `datetime` value to a `TIME` value.

Example

```
OceanBase (root@test)> SELECT TIME('2003-12-31 01:02:03');
+-----------------------------+
| TIME('2003-12-31 01:02:03') |
+-----------------------------+
| 01:02:03.000000             |
+-----------------------------+
1 row in set (0.01 sec)
```

TIME_TO_USEC

Declaration

```
TIME_TO_USEC(date)
```

Description

This function converts a `date` value to the number of microseconds that have elapsed since `1970-01-01 00:00:00.000000` based on the time zone.

A `date` value can contain the date or both the date and the time.

Example

```
OceanBase (root@test)> SELECT TIME_TO_USEC('2014-03-25'), TIME_TO_USEC(NOW());
+----------------------------+---------------------+
| TIME_TO_USEC('2014-03-25') | TIME_TO_USEC(NOW()) |
+----------------------------+---------------------+
|           1395676800000000 |    1525528100000000 |
+----------------------------+---------------------+
1 row in set (0.01 sec)
```

TIMEDIFF

Declaration

```
TIMEDIFF(date1, date2)
```

Description

This function returns a `TIME` value that represents the time interval between two date and time values.

Example

```
OceanBase (root@test)> SELECT
    -> TIMEDIFF('2015-06-06 12:12:12', '2014-06-05 11:11:11'),
    -> TIMEDIFF('2015-06-06 12:12:12', '2015-06-05 11:11:11')
    -> \G
*************************** 1. row ***************************
TIMEDIFF('2015-06-06 12:12:12', '2014-06-05 11:11:11'): 838:59:59
TIMEDIFF('2015-06-06 12:12:12', '2015-06-05 11:11:11'): 25:01:01
1 row in set (0.00 sec)
```

TIMESTAMPDIFF

Declaration

```
TIMESTAMPDIFF(unit, date1, date2)
```

Description

This function returns the interval between two date and time values in the specified `unit`. `unit` can only be one of the independent units provided in the description of the `DATE_ADD()` function.

Example

```
OceanBase (root@test)> SELECT
    -> TIMESTAMPDIFF(SECOND, NOW(), '2011-01-01 11:11:11'),
    -> TIMESTAMPDIFF(DAY, '2011-01-01 11:11:11', NOW())
    -> \G
*************************** 1. row ***************************
TIMESTAMPDIFF(SECOND, NOW(), '2011-01-01 11:11:11'): -231677498
   TIMESTAMPDIFF(DAY, '2011-01-01 11:11:11', NOW()): 2681
1 row in set (0.00 sec)
```

TIMESTAMPADD

Declaration

```
TIMESTAMPADD(unit, interval_expr, date)
```

Description

This function performs arithmetic calculations on date and time values.

This function performs the same operation as `DATE_ADD()`. However, the `unit` value must be an independent unit.

Example

```
OceanBase (root@test)> SELECT
    -> TIMESTAMPADD(DAY, -5, '2010-01-01 00:00:00'),
    -> DATE_ADD('2010-01-01 00:00:00', INTERVAL -5 DAY)
    -> \G
*************************** 1. row ***************************
    TIMESTAMPADD(DAY, -5, '2010-01-01 00:00:00'): 2009-12-27 00:00:00
DATE_ADD('2010-01-01 00:00:00', INTERVAL -5 DAY): 2009-12-27 00:00:00
1 row in set (0.01 sec)
```

TO_DAYS

Declaration

```
TO_DAYS(date)
```

Description

This function returns the number of days based on the specified `date` value. The number of days indicates the number of days that have elapsed since `0000-01-01`.

Example

```
OceanBase (root@test)> SELECT TO_DAYS('2015-11-04'), TO_DAYS('20151104');
+-----------------------+---------------------+
| TO_DAYS('2015-11-04') | TO_DAYS('20151104') |
+-----------------------+---------------------+
|                736271 |              736271 |
+-----------------------+---------------------+
1 row in set (0.01 sec)
```

USEC_TO_TIME

Declaration

```
USEC_TO_TIME(usec)
```

Description

This function converts a `usec` value to a `TIMESTAMP` value.

`usec` is the number of microseconds that have elapsed since `1970-01-01 00:00:00.000000` based on the time zone.

Example

```
OceanBase (root@test)> SELECT USEC_TO_TIME(1);
+----------------------------+
| USEC_TO_TIME(1)            |
+----------------------------+
| 1970-01-01 08:00:00.000001 |
+----------------------------+
1 row in set (0.00 sec)
```

UNIX_TIMESTAMP

Declaration

```
UNIX_TIMESTAMP()
UNIX_TIMESTAMP(date)
```

Description

- If you do not specify the `date` parameter, the function returns the number of seconds that have elapsed since 1970-01-01 00:00:00 based on the time zone.
- If you specify the `date` parameter, the function returns the number of seconds between the specified time and 1970-01-01 00:00:00 based on the time zone.

Example

```
OceanBase (root@test)> SELECT UNIX_TIMESTAMP(), TIME_TO_USEC(NOW());
+------------------+---------------------+
| UNIX_TIMESTAMP() | TIME_TO_USEC(NOW()) |
+------------------+---------------------+
|       1525570561 |    1525570561000000 |
+------------------+---------------------+
1 row in set (0.01 sec)

OceanBase (root@test)> SELECT UNIX_TIMESTAMP('1997-10-04 22:23:00');
+---------------------------------------+
| UNIX_TIMESTAMP('1997-10-04 22:23:00') |
+---------------------------------------+
|                             875974980 |
+---------------------------------------+
1 row in set (0.01 sec)
```

UTC_TIMESTAMP

Declaration

```
UTC_TIMESTAMP()
```

Description

This function returns the current UTC time.

Example

```
OceanBase (root@test)> SELECT UTC_TIMESTAMP();
+---------------------+
| UTC_TIMESTAMP()     |
+---------------------+
| 2018-05-06 01:38:32 |
+---------------------+
1 row in set (0.01 sec)
```

YEAR

Declaration

```
YEAR(date)
```

Description

This function returns the year of the `date` value.

Example

```
OceanBase (root@test)> SELECT YEAR('1987-01-01');
+--------------------+
| YEAR('1987-01-01') |
+--------------------+
|               1987 |
+--------------------+
1 row in set (0.00 sec)
```

String functions

CONCAT

Declaration

```
CONCAT(str1, .., strN)
```

Description

This function concatenates multiple strings into one string. If an argument is `NULL`, the function returns `NULL`.

Example

```
OceanBase (root@test)> SELECT
    -> CONCAT('test','OceanBase', '1.0'),
    -> CONCAT('test','OceanBase', NULL)
    -> \G
*************************** 1. row ***************************
CONCAT('test','OceanBase', '1.0'): testOceanBase1.0
 CONCAT('test','OceanBase', NULL): NULL
1 row in set (0.01 sec)
```

CONCAT_WS

Declaration

```
CONCAT_WS(separator, str1, .., strN)
```

Description

This function concatenates multiple strings into one string, and separates adjacent strings with a `separator`. If an argument is `NULL`, the function ignores the `NULL` value.

Example

```
OceanBase (root@test)> SELECT
    -> CONCAT_WS('_', 'First', 'Second'),
    -> CONCAT_WS('_', 'First', NULL, 'Second')
    -> \G
*************************** 1. row ***************************
      CONCAT_WS('_', 'First', 'Second'): First_Second
CONCAT_WS('_', 'First', NULL, 'Second'): First_Second
1 row in set (0.00 sec)
```

SUBSTR

Declaration

```
SUBSTR(str, pos)
SUBSTR(str, pos, len)
SUBSTR(str FROM pos)
SUBSTR(str FROM pos FOR len)
```

Description

This function returns a substring of length `len` by starting at position `pos` of string `str`. If an argument is `NULL`, the function returns `NULL`.

- If you do not specify `len`, the returned substring starts from `pos` to the end of `str`.
- If `pos` is negative, the start position is determined by counting backward from the end of `str`.
- If `len` is less than or equal to 0, or `pos` is an invalid start position, an empty string is returned.

Example

```
OceanBase (root@test)> SELECT
    -> SUBSTR('abcdefg', 3),
    -> SUBSTR('abcdefg', 3, 2),
    -> SUBSTR('abcdefg', -3),
    -> SUBSTR('abcdefg', 3, -2),
    -> SUBSTR('abcdefg' from -4 for 2)
    -> \G
*************************** 1. row ***************************
           SUBSTR('abcdefg', 3): cdefg
        SUBSTR('abcdefg', 3, 2): cd
          SUBSTR('abcdefg', -3): efg
       SUBSTR('abcdefg', 3, -2):
SUBSTR('abcdefg' from -4 for 2): de
1 row in set (0.01 sec)
```

SUBSTRING

Declaration

```
SUBSTRING(str, pos)
SUBSTRING(str, pos, len)
SUBSTRING(str FROM pos)
SUBSTRING(str FROM pos FOR len)
```

Description

This function performs the same operation as `SUBSTR`.

TRIM

Declaration

```
TRIM([[{BOTH | LEADING | TRAILING}] [remstr] FROM] str)
```

Description

This function removes prefixes, suffixes, or both from a string. By default, the value is `BOTH`. If an argument is `NULL`, the function returns `NULL`.

Example

```
OceanBase (root@test)> SELECT
    -> TRIM(' bar '),
    -> TRIM(LEADING 'x' FROM 'xxxbarxxx'),
    -> TRIM(BOTH 'x' FROM 'xxxbarxxx'),
    -> TRIM(TRAILING 'x' FROM 'xxxbarxxx')
    -> \G
*************************** 1. row ***************************
                    TRIM(' bar '): bar
 TRIM(LEADING 'x' FROM 'xxxbarxxx'): barxxx
    TRIM(BOTH 'x' FROM 'xxxbarxxx'): bar
TRIM(TRAILING 'x' FROM 'xxxbarxxx'): xxxbar
1 row in set (0.01 sec)
```

LTRIM

Declaration

```
LTRIM(str)
```

Description

This function removes the spaces to the left of the string.

RTRIM

Declaration

```
RTRIM(str)
```

Description

This function removes the spaces to the right of the string.

ASCII

Declaration

```
ASCII(str)
```

Description

This function returns the ASCII code of the leftmost character of a string.

ORD

Declaration

```
ORD(str)
```

Description

This function returns the character code of the leftmost character of a string. If the leftmost character is a multibyte character, the function uses the following rules to calculate the code:

```
(1st byte code)
+ (2nd byte code * 256)
+ (3rd byte code * 256^2) ...
```

Example

```
OceanBase (root@test)> SELECT ORD('China');
+---------------+
| ORD('China') |
+---------------+
|      14989485 |
+---------------+
1 row in set (0.01 sec)
```

LENGTH

Declaration

```
LENGTH(str)
```

Description

This function returns the length of `str` in bytes.

Example

```
OceanBase (root@test)> SELECT LENGTH('China'), LENGTH('hello');
+------------------+-----------------+
| LENGTH('China') | LENGTH('hello') |
+------------------+-----------------+
|                6 |               5 |
+------------------+-----------------+
1 row in set (0.01 sec)
```

CHAR_LENGTH

Declaration

```
CHAR_LENGTH(str)
```

Description

This function returns the number of characters in a string.

Example

```
OceanBase (root@test)> SELECT CHAR_LENGTH('China'), CHAR_LENGTH('hello');
+-----------------------+----------------------+
| CHAR_LENGTH('China') | CHAR_LENGTH('hello') |
+-----------------------+----------------------+
|                     2 |                    5 |
+-----------------------+----------------------+
1 row in set (0.00 sec)
```

UPPER

Declaration

```
UPPER(str)
```

Description

This function converts lowercase letters in a string to uppercase letters.

Example

```
OceanBase (root@test)> SELECT UPPER('Hello, OceanBase!') ;
+------------------------------+
| UPPER('Hello, OceanBase!') |
+------------------------------+
| HELLO, OCEANBASE!          |
+------------------------------+
1 row in set (0.01 sec)
```

LOWER

Declaration

```
LOWER(str)
```

Description

This function converts uppercase letters in a string to lowercase letters.

Example

```
OceanBase (root@test)> SELECT LOWER('Hello, OceanBase!') ;
+------------------------------+
| LOWER('Hello, OceanBase!') |
+------------------------------+
| hello, oceanbase!          |
+------------------------------+
1 row in set (0.01 sec)
```

HEX

Declaration

```
HEX(str)
```

Description

This function converts a number or a string into a hexadecimal string.

Example

```
OceanBase (root@test)> SELECT HEX(255), HEX('abc');
+----------+------------+
| HEX(255) | HEX('abc') |
+----------+------------+
| FF       | 616263     |
+----------+------------+
1 row in set (0.00 sec)
```

UNHEX

Declaration

```
UNHEX(str)
```

Description

This function converts a hexadecimal string to a normal string.

Example

```
OceanBase (root@test)> SELECT UNHEX('4f6365616e42617365');
+-----------------------------+
| UNHEX('4f6365616e42617365') |
+-----------------------------+
| OceanBase                   |
+-----------------------------+
1 row in set (0.00 sec)
```

MD5

Declaration

```
MD5(str)
```

Description

This function returns the MD5 value of a string.

Example

```
OceanBase (root@test)> SELECT MD5(1);
+----------------------------------+
| MD5(1)                           |
+----------------------------------+
| c4ca4238a0b923820dcc509a6f75849b |
+----------------------------------+
1 row in set (0.00 sec)
```

INT2IP

Declaration

```
INT2IP(int_value)
```

Description

This function converts an integer to an IP address.

Example

```
OceanBase (root@test)> SELECT
    -> INT2IP(16777216),
    -> HEX(16777216),
    -> INT2IP(1)
    -> \G
*************************** 1. row ***************************
INT2IP(16777216): 1.0.0.0
   HEX(16777216): 1000000
       INT2IP(1): 0.0.0.1
1 row in set (0.01 sec)
```

IP2INT

Declaration

```
IP2INT('ip_addr')
```

Description

This function converts an IP address to an integer.

Example

```
OceanBase (root@test)> SELECT
    -> IP2INT('0.0.0.1'),
    -> HEX(IP2INT('0.0.0.1')),
    -> HEX(IP2INT('1.0.0.0'))
    -> \G
*************************** 1. row ***************************
     IP2INT('0.0.0.1'): 1
HEX(IP2INT('0.0.0.1')): 1
HEX(IP2INT('1.0.0.0')): 1000000
1 row in set (0.01 sec)
```

LIKE

Declaration

```
str1 [NOT] LIKE str2 [ESCAPE str3]
```

Description

This function compares strings based on wildcards. If an argument is `NULL`, the function returns `NULL`.

You can use the following wildcards:

- `%`: matches a string. No limits apply to the string length.
- `_`: matches a single character.

ESCAPE is used to define an escape character. If str3 is included in str2, the characters after str3 are processed as common characters during the match.

Example

```
OceanBase (root@test)> SELECT 'ab%' LIKE 'abc%' ESCAPE 'c';
+------------------------------+
| 'ab%' LIKE 'abc%' ESCAPE 'c' |
+------------------------------+
|                            1 |
+------------------------------+
1 row in set (0.01 sec)
```

REGEXP

Declaration

```
str [NOT] REGEXP | RLIKE pat
```

Description

This function performs pattern matching based on regular expressions. If an argument is `NULL`, the function returns `NULL`.

Example

```
OceanBase (root@test)> SELECT
    -> 1234 REGEXP 1,
    -> 'hello' RLIKE 'h%'
    -> \G
*************************** 1. row ***************************
     1234 REGEXP 1: 1
'hello' RLIKE 'h%': 0
1 row in set (0.01 sec)
```

REPEAT

Declaration

```
REPEAT(str, count)
```

Description

This function returns a string by repeating `str` for `count` times. If the `count` value is less than or equal to 0, an empty string is returned. If an argument is `NULL`, the function returns `NULL`.

Example

```
OceanBase (root@test)> SELECT
    -> REPEAT('1', -1),
    -> REPEAT(null, null),
    -> REPEAT('Abc', 4)
    -> \G
*************************** 1. row ***************************
   REPEAT('1', -1):
REPEAT(null, null): NULL
  REPEAT('Abc', 4): AbcAbcAbcAbc
1 row in set (0.01 sec)
```

SPACE

Declaration

```
SPACE(N)
```

Description

This function returns a string that contains `N` spaces.

SUBSTRING_INDEX

Declaration

```
SUBSTRING_INDEX(str, delim, count)
```

Description

This function returns the substring from string str before count occurrences of the delimiter delim. If the count value is positive, the function returns everything to the left of the final delimiter (counting from the left). If the count value is negative, the function returns everything to the right of the final delimiter (counting from the right). If an argument is NULL, the function returns NULL. If the str or delim value is an empty string, the function returns an empty string. If the count value is 0, the function returns an empty string.

str, delim, and count support implicit conversion between numbers and strings.

Example

```
Oceanbase>select substring_index('abcdabc', 'abc', 0), substring_index('abcdabc', 'abc', 1), substrin
g_index('abcdabc', 'abc', 2), substring_index('abcdabc', 'abc', 3), substring_index('abcdabc', 'abc',
-1), substring_index('abcdabc', 'abc', -2), substring_index('abcdabc', 'abc', -3)\G;
    * 1. row *
     substring_index('abcdabc', 'abc', 0):
     substring_index('abcdabc', 'abc', 1):
     substring_index('abcdabc', 'abc', 2): abcd
     substring_index('abcdabc', 'abc', 3): abcdabc
    substring_index('abcdabc', 'abc', -1):
    substring_index('abcdabc', 'abc', -2): dabc
    substring_index('abcdabc', 'abc', -3): abcdabc
    1 row in set (0.00 sec)
```

LOCATE

Declaration

```
LOCATE(substr,str) , LOCATE(substr,str,pos)
```

Description

The first syntax returns the position of the first occurrence of the substring substr in the string str. The second syntax returns the position of the first occurrence of the substring substr in the string str, which starts at the position pos. If substr is not included in str, the function returns 0.

Example

```
Oceanbase>SELECT LOCATE('bar', 'foobarbar');
        -> 4
Oceanbase>SELECT LOCATE('xbar', 'foobar');
        -> 0
Oceanbase>SELECT LOCATE('bar', 'foobarbar',5);
        -> 7
```

POSITION

Declaration

```
POSITION(substr IN str)
```

Description

This function performs the same operation as LOCATE.

INSTR

Declaration

```
INSTR(str,substr)
```

Description

This function returns the position of the first occurrence of the substring in str. This is the same as the two-parameter form of LOCATE(), except that the order of the parameters is reversed.

Example

```
Oceanbase>SELECT INSTR('foobarbar', 'bar');
        -> 4
Oceanbase>SELECT INSTR('xbar', 'foobar');
        -> 0
```

REPLACE

Declaration

```
REPLACE(str, from_str, to_str)
```

Description

This function returns the str string after all occurrences of from_str are replaced by to_str.

Example

```
Oceanbase>SELECT REPLACE('abc.efg.gpg.nowdew.abc.dabc.e', 'abc.', 'www');
+-----------------------------------------------------------+
| REPLACE('abc.efg.gpg.nowdew.abc.dabc.e', 'abc.', 'www') |
+-----------------------------------------------------------+
| wwwefg.gpg.nowdew.wwwdwwwe                                |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

FIELD

Declaration

```
FIELD(str,str1,str2,str3,...)
```

Description

This function returns the index position of the str string in the list of strings str1, str2, str3,... The sequence numbers for index positions start from 1. If str is not found, the function returns 0.

If all arguments in the FIELD() function are strings, the function compares the arguments based on the strings. If all arguments are numbers, the function compares the arguments based on the numbers. Otherwise, the function compares the arguments as values of the double data type.

If the str value is NULL, the function returns 0. This is because NULL cannot be compared with the other values. FIELD() is a supplement to ELT().

Example

```
Oceanbase>select field('abc','abc1','abc2','abc','abc4','abc'), field(NULL, 'null1', NULL);
+-----------------------------------------------+-----------------------------+
| field('abc','abc1','abc2','abc','abc4','abc') | field(NULL, 'null1', NULL) |
+-----------------------------------------------+-----------------------------+
|                                             3 |                           0 |
+-----------------------------------------------+-----------------------------+
1 row in set (0.00 sec)
```

ELT

Declaration

```
ELT(N, str1, str2, str3,...)
```

Description

This function returns the Nth element of a list of strings. For example, if you set N to 1, the function returns the string str1. If you set N to 2, the function returns the string str2. If N is less than 1 or greater than the number of the parameters, the function returns NULL. ELT() is a supplement to FIELD().

Example

```
Oceanbase>select elt(3, 'abc1', 'abc2', 'abc', 'abc4', 'abc'), elt(0, 'null1', NULL);
+-----------------------------------------------+-----------------------+
| elt(3, 'abc1', 'abc2', 'abc', 'abc4', 'abc') | elt(0, 'null1', NULL) |
+-----------------------------------------------+-----------------------+
| abc                                           | NULL                  |
+-----------------------------------------------+-----------------------+
1 row in set (0.00 sec)
```

INSERT

Declaration

```
INSERT (str1,pos,len,str2)
```

Description

This function replaces a specified substring of a string with a new substring. The str1 parameter specifies the source string and the str2 parameter specifies the new substring. The pos parameter specifies the start position of the original substring and the len parameter specifies the length of the original substring. If the pos value is greater than the length of the source string, the function returns the source string. If the len value is greater than the length of the str1 or str2 string, the function replaces the original substring that starts at the pos position. If an argument is NULL, the function returns NULL. This function supports multibyte characters.

- *The values of str1 and str2 must be strings. The values of pos and len must be integers. If an argument is NULL, the function returns NULL.*
- *The text characters in str1 and str2 are identified as byte streams.*
- *If the pos value is negative or greater than the length of the string str1, the function returns the string str1.*
- If the len value is less than 0 or greater than the length of the str value, the function returns a string that consists of the string str2 and a substring of the string str1. The substring starts from the first character of the string str1 value and ends at the position that is specified by the pos parameter.

Example

```
Oceanbase>select insert('Quadratic',-2,100,'What'), insert('Quadratic',7,3,'What'),
    -> insert('Quadratic',-1,3,'What'), insert('Quadratic',10,3,'What'), insert('Quadratic',5,-1,''),
    -> insert('Quadratic',7,-1,'What')\G;
* 1. row *
insert('Quadratic',-2,100,'What'): Quadratic
   insert('Quadratic',7,3,'What'): QuadraWhat
  insert('Quadratic',-1,3,'What'): Quadratic
  insert('Quadratic',10,3,'What'): Quadratic
      insert('Quadratic',5,-1,''): Quad
  insert('Quadratic',7,-1,'What'): QuadraWhat
1 row in set (0.01 sec)
```

LPAD

Declaration

```
LPAD(str, len, padstr)
```

Description

This function adds the padstr padding string to the left of the str string until the final string reaches the len value. If the str value is longer than the len value, str is truncated.

Example

```
OceanBase > SELECT LPAD('hi',4,'??') ;
+-------------------+
| LPAD('hi',4,'??') |
+-------------------+
| ?? hi             |
+-------------------+
1 row in set (0.01 sec)

OceanBase > SELECT LPAD('hi',1,'??') ;
+-------------------+
| LPAD('hi',1,'??') |
+-------------------+
| h                 |
+-------------------+
1 row in set (0.00 sec)
```

RPAD

Declaration

```
RPAD(str, len, padstr)
```

Description

This function adds the padstr padding string to the right of the str string until the final string reaches the len value. If the str value is longer than the len value, str is truncated.

Example

```
OceanBase (root@test)> SELECT RPAD('hi',4,'??') ;
+-------------------+
| RPAD('hi',4,'??') |
+-------------------+
| hi??              |
+-------------------+
1 row in set (0.00 sec)

OceanBase (root@test)> SELECT RPAD('hi',1,'??') ;
+-------------------+
| RPAD('hi',1,'??') |
+-------------------+
| h                 |
+-------------------+
1 row in set (0.00 sec)
```

UUID

Declaration

```
uuid()
```

Description

This function generates a globally unique identifier (ID).

Example

```
OceanBase (root@test)> select uuid();
+--------------------------------------+
| uuid()                               |
+--------------------------------------+
| f756a1f6-4de6-11e8-90af-90b11c53e421 |
+--------------------------------------+
1 row in set (0.00 sec)
```

BIN

Declaration

```
bin(N)
```

Description

This function returns the binary form of the number N.

Example

```
OceanBase > SELECT BIN(12);
+----------+
| BIN(12) |
+----------+
| 1100    |
+----------+
1 row in set (0.00 sec)
```

QUOTE

Declaration

```
quote(str)
```

Description

This function quotes a string to produce a result that can be used as a properly escaped data value in an SQL statement. This function returns a string that is enclosed by single quotation marks ('), and each single quotation marks ('), backslash (\), ASCII NUL, and Ctrl+Z is preceded with a backslash (\). If the argument is NULL, the function returns a word NULL that is not enclosed by single quotation marks (').

Example

```
OceanBase > SELECT QUOTE('Don\'t!') ;
+------------------+
| QUOTE('Don\'t!') |
+------------------+
| 'Don\'t!'        |
+------------------+
1 row in set (0.00 sec)

OceanBase > SELECT QUOTE(NULL);
+-------------+
| QUOTE(NULL) |
+-------------+
| NULL        |
+-------------+
1 row in set (0.00 sec)
```

REGEXP_SUBSTR

Declaration

```
regexp_substr(str,pattern,[position[,occurrence[,match_param[,subexpr]]]])
```

Description

This function searches for a substring that matches the regular expression pattern in the str string. If the substring does not exist, the function returns NULL. This function supports multibyte characters. Except for match_param, if an argument is NULL, the function returns NULL.

- str is the string to be searched. Multibyte characters are supported.
- pattern is the regular expression, and the regular expression rules are compatible with MySQL.
- position is an optional parameter that specifies where to begin the search. The position parameter must be a positive integer. An error is reported if the value is less than or equal to 0. If the input value is NULL, the function returns NULL. By default, the value is 1, which means that the search starts from the first character.
- occurrence is an optional parameter. This parameter specifies the number of times the string matches the regular expression. The occurrence parameter must be an integer that is greater than 0. An error is reported if the value is less than or equal to 0. If the input value is NULL, the function returns NULL. The default value is 1, which means that the function searches for the first occurrence of the pattern.
- match_param is an optional parameter that specifies the string type. This parameter supports only two characters: i and c. i specifies case-insensitive matching. c specifies case-sensitive matching. If you specify other characters, an error occurs. The default value is determined by the character set of str. The default value is used if match_param is NULL.
- subexpr is an optional parameter. This parameter specifies which group of the regular expression is used to match the string. subexpr must be an integer greater than or equal to 0. An error is reported if the value is smaller than 0. The default value is 0, which means that the function returns a substring that satisfies the entire pattern.

Example

```
oceanbase> select regexp_substr('I have 2 apples and 100 bucks!', '[[:blank:]][[:alnum:]]*', 1, 1) fr
om dual;
+-----------------------------------------------------------------------------------+
| regexp_substr('I have 2 apples and 100 bucks!', '[[:blank:]][[:alnum:]]*', 1, 1) |
+-----------------------------------------------------------------------------------+
|  have                                                                             |
+-----------------------------------------------------------------------------------+
1 row in set (0.00 sec)

oceanbase> select regexp_substr('foothebar', 'foo(.*)(bar)', 1, 1, 'c', 1) from dual;
+-----------------------------------------------------------+
| regexp_substr('foothebar', 'foo(.*)(bar)', 1, 1, 'c', 1) |
+-----------------------------------------------------------+
| the                                                       |
+-----------------------------------------------------------+
1 row in set (0.01 sec)
```

Type conversion functions

CAST

Declaration

```
CAST(expr AS type)
```

Description

This function explicitly converts an expression of one data type to another data type.

This function converts the value of expr to a new data type.

Parameters:

- expr specifies a valid SQL expression.
- AS separates two parameters. The parameter before AS specifies the data to be processed. The parameter after AS specifies the destination data type.
- type specifies a valid data type supported by ApsaraDB for OceanBase. Valid values:
  - DATE
  - DATETIME
  - DECIMAL
  - SIGNED [INTEGER]
  - TIME
  - UNSIGNED [INTEGER]

The CAST function for type conversion is applicable if one of the following conditions is met:

- The data types of the two expressions are the same.
- The data types of the two expressions can be converted in an implicit way.
- The data types must be explicitly converted.

If you attempt to perform an invalid conversion, ApsaraDB for OceanBase returns an error message.

If the length of a data type is not specified, the system uses the maximum length that is supported for the data type in ApsaraDB for OceanBase. For example, the maximum length for the VARCHAR data type is 262,143 bytes. The maximum length for a numeric data type is 65 bits for floating-point numbers.

You can use the CAST function to convert signed and unsigned 64-bit values. If you use a numeric operator such as a plus sign (+) and one of the operands is an unsigned integer, the function returns an unsigned value. To override a numeric operator, use the SIGNED or UNSIGNED cast operator to cast a value to a signed or unsigned 64-bit integer.

If an operand is a floating-point value, the result is a floating-point value.

Example

```
Oceanbase>SELECT CAST(123 AS BOOL);
+-------------------+
| CAST(123 AS bool) |
+-------------------+
|                 1 |
+-------------------+
1 row in set (0.00 sec)

Oceanbase>select cast(1-2 as unsigned), cast(cast(1-2 as unsigned) as signed);
+-----------------------+---------------------------------------+
| cast(1-2 as unsigned) | cast(cast(1-2 as unsigned) as signed) |
+-----------------------+---------------------------------------+
|  18446744073709551615 |                                    -1 |
+-----------------------+---------------------------------------+
1 row in set (0.00 sec)

Oceanbase>SELECT CAST(1 AS UNSIGNED) - 2.0;
+----------------------------+
| CAST(1 AS UNSIGNED) - 2.0 |
+----------------------------+
|                      -1.0 |
+----------------------------+
1 row in set (0.00 sec)

Oceanbase>select cast(0 as date);
+-----------------+
| cast(0 as date) |
+-----------------+
| 0000-00-00      |
+-----------------+
1 row in set (0.00 sec)
```

Mathematical functions

ROUND

Declaration

```
ROUND(X), ROUND(X,D)
```

Description

This function returns a number that is rounded to the specified length or precision.

This function rounds the X value to the nearest integer. If you specify two parameters for the function, the function rounds X to D decimal places. The Dth digit is obtained by rounding. To keep D digits of the X value to the left of the decimal point, set D to a negative value.

The data type of the returned value is the same as that of the first argument. You can assume that the value is an integer, a double-precision floating-point number, or a decimal value. This means that the returned result for an integer argument is also an integer. No fractional part is included in the returned result.

- For exact-valued numbers, ROUND() uses the rules of rounding up or rounding to the nearest number. A value that has a fractional part of .5 or greater is rounded up to the next integer if positive or down to the next integer if negative. In other words, the value is rounded away from zero on the number axis. A value that has a fractional part less than .5 is rounded down to the next integer if positive or up to the next integer if negative.
- For approximate numbers, ROUND() follows the bankers rounding rule. If a value has a fractional part, the function rounds the value to the nearest even integer.

Example

```
Oceanbase>select round(2.15,2);
+---------------+
| round(2.15,2) |
+---------------+
|          2.15 |
+---------------+
1 row in set (0.00 sec)

Oceanbase>select round(2555e-2,1);
+------------------+
| round(2555e-2,1) |
+------------------+
|             25.6 |
+------------------+
1 row in set (0.01 sec)

Oceanbase>select round(25e-1), round(25.3e-1),round(35e-1);
+--------------+----------------+--------------+
| round(25e-1) | round(25.3e-1) | round(35e-1) |
+--------------+----------------+--------------+
|            3 |              3 |            4 |
+--------------+----------------+--------------+
1 row in set (0.00 sec)
```

CEIL

Declaration

```
CEIL(expr)
```

Description

This function returns the smallest integer value that is no less than the specified expression.

This function supports comparison operations. The comparison result is a BOOLEAN value. The BOOLEAN value is converted to a numeric value: 1 for TRUE or 0 for FALSE.

If the value of expr is NULL, the function returns NULL.

If you specify a string of numbers, the function automatically converts the string to a numeric value.

The returned value is converted to a BIGINT number.

Example

```
Oceanbase>select ceil(1.2), ceil(-1.2), ceil(1+1.5), ceil(1=1),ceil(1<1),ceil(null);
+-----------+------------+-------------+-----------+-----------+------------+
| ceil(1.2) | ceil(-1.2) | ceil(1+1.5) | ceil(1=1) | ceil(1<1) | ceil(null) |
+-----------+------------+-------------+-----------+-----------+------------+
|         2 |         -1 |           3 |         1 |         0 |       NULL |
+-----------+------------+-------------+-----------+-----------+------------+
1 row in set (0.00 sec)

Oceanbase>select ceil(name);
ERROR 1166 (42703): Unkown column name 'name'

Oceanbase>select ceil('2');
+------------+
| ceil('2') |
+------------+
|         2 |
+------------+
1 row in set (0.00 sec)
```

FLOOR

Declaration

```
FLOOR(expr)
```

Description

This function is similar to CEIL(expr) and returns the largest integer value that is less than or equal to the specified expression.

The function supports comparison operations. The comparison result is a BOOLEAN value. The BOOLEAN value is converted to a numeric value: 1 for TRUE or 0 for FALSE.

If the value of expr is NULL, the function returns NULL.

If you specify a string of numbers, the function automatically converts the string to a numeric value.

The returned value is converted to a BIGINT number.

Example

```
Oceanbase>select floor(1.2), floor(-1.2), floor(1+1.5), floor(1=1),floor(1<1),floor(null);
+------------+-------------+--------------+------------+------------+-------------+
| floor(1.2) | floor(-1.2) | floor(1+1.5) | floor(1=1) | floor(1<1) | floor(null) |
+------------+-------------+--------------+------------+------------+-------------+
|          1 |          -2 |            2 |          1 |          0 |        NULL |
+------------+-------------+--------------+------------+------------+-------------+
1 row in set (0.00 sec)

Oceanbase>select floor(name);
ERROR 1166 (42703): Unkown column name 'name'

Oceanbase>select floor('2');
+-------------+
| floor('2') |
+-------------+
|          2 |
+-------------+
1 row in set (0.00 sec)
```

ABS

Declaration

```
ABS(expr)
```

Description

This function returns the absolute value for a specified numeric expression. The data type of the returned value is the same as that of the expression value.

The function supports comparison operations. The comparison result is a BOOLEAN value. The BOOLEAN value is converted to a numeric value: 1 for TRUE or 0 for FALSE.

If the value of expr is NULL, the function returns NULL.

If you specify a string of numbers, the function automatically converts the string to a numeric value.

The returned value is converted to a BIGINT number.

Example

```
Oceanbase>select abs(5), abs(-5.777), abs(0), abs(1/2), abs(1-5);
+--------+-------------+--------+----------+----------+
| abs(5) | abs(-5.777) | abs(0) | abs(1/2) | abs(1-5) |
+--------+-------------+--------+----------+----------+
|      5 |       5.777 |      0 |   0.5000 |        4 |
+--------+-------------+--------+----------+----------+
1 row in set (0.00 sec)
```

NEG

Declaration

```
NEG(expr)
```

Description

This function is a negation function that subtracts a specified operand from zero and returns the final result.

The function supports comparison operations. The comparison result is a BOOLEAN value. The BOOLEAN value is converted to a numeric value: 1 for TRUE or 0 for FALSE.

Example

```
Oceanbase>select neg(1), neg(1+1), neg(2*3), neg(1=1), neg(5<1);
+--------+----------+----------+----------+----------+
| neg(1) | neg(1+1) | neg(2*3) | neg(1=1) | neg(5<1) |
+--------+----------+----------+----------+----------+
|     -1 |       -2 |       -6 |       -1 |        0 |
+--------+----------+----------+----------+----------+
1 row in set (0.01 sec)
```

SIGN

Declaration

```
SIGN(X)
```

Description

This function returns the sign of a specified number. The function returns -1, 0, or 1 based on whether the specified X value is negative, zero, or positive.

The function supports comparison operations. The comparison result is a BOOLEAN value. The BOOLEAN value is converted to a numeric value: 1 for TRUE or 0 for FALSE.

If the value of X is NULL, the function returns NULL.

The function supports floating-point numbers and hexadecimal numbers.

Example

```
Oceanbase>SELECT SIGN(-32), SIGN(0), SIGN(234);
+-----------+---------+-----------+
| SIGN(-32) | SIGN(0) | SIGN(234) |
+-----------+---------+-----------+
|        -1 |       0 |         1 |
+-----------+---------+-----------+
1 row in set (0.01 sec)

Oceanbase>select sign(null),sign(false),sign(0x01);
+------------+-------------+------------+
| sign(null) | sign(false) | sign(0x01) |
+------------+-------------+------------+
|       NULL |           0 |          1 |
+------------+-------------+------------+
1 row in set (0.00 sec)
```

CONV

Declaration

```
CONV(N, from_base, to_base)
```

Description

This function converts a number from one base to another base. The function converts a number from the from_base base to the to_base base. The returned result is a string. The value of the N input parameter can be an integer or a string. The minimum base is 2 and the maximum base is 36. If the to_base value is a negative number, N is processed as a signed number. Otherwise, N is processed as an unsigned number. If the from_base value is a negative number, the value is processed as an integer and the value sign is ignored. The N parameter must be an integer or a string. The from_base and to_base parameters must be decimal integers within the valid value range: the union of [-36,-2] and [2,36].

Invalid input values result in errors. In the following scenarios, the input values are invalid:

- *The value of the from_base or to_base parameter is an invalid decimal integer value.*
- *The value of the from_base or to_base parameter does not fall within the valid value range: the union of [-36,-2] and [2,36].*
- N is an invalid numeric value. For example, N falls out of the following character range: 0 to 9, a to z, or A to Z.
- N falls out of the valid range of the from_base values. For example, the from_base value is 2, whereas N is 3.
- N falls out of the valid range of 64-bit integers. The valid range is [-9223372036854775807,9223372036854775807].

Example

```
Oceanbase>select conv(9223372036854775807,10,2);
+----------------------------------------------------------------+
| conv(9223372036854775807,10,2)                                 |
+----------------------------------------------------------------+
| 1111111111111111111111111111111111111111111111111111111111111111 |
+----------------------------------------------------------------+
1 row in set (0.00 sec)

Oceanbase>select conv('-acc',21,-7);
+--------------------+
| conv('-acc',21,-7) |
+--------------------+
| -16425             |
+--------------------+
1 row in set (0.00 sec)
```

MOD

Declaration

```
MOD(N,M)
```

Description

This function returns the remainder after a number is divided by another number. MOD(N,M), N % M, and N MOD M perform the same operations.

You can use the MOD() function to perform calculations on numbers that have fractional parts. This function returns the exact remainder after a number is divided by another number.

If one of the N and M values is NULL, the function returns NULL. If M is 0, the function returns NULL.

Example

```
Oceanbase>select mod(29,19), 29 mod 19, 29 % 19;
+------------+-----------+----------+
| mod(29,19) | 29 mod 19 | 29 % 19 |
+------------+-----------+----------+
|         10 |        10 |      10 |
+------------+-----------+----------+
1 row in set (0.00 sec)
```

POW

Declaration

```
POW(X,Y)
```

Description

This function raises X to the power of Y.

If one of the X and Y values is NULL, the function returns NULL.

Example

```
Oceanbase>select pow(4,2), pow(4,-2), pow(1,null);
+----------+-----------+-------------+
| pow(4,2) | pow(4,-2) | pow(1,null) |
+----------+-----------+-------------+
|       16 |    0.0625 |        NULL |
+----------+-----------+-------------+
1 row in set (0.00 sec)
```

POWER

Declaration

```
POWER(X,Y)
```

Description

POWER(X,Y) and POW(X,Y) are identical functions.

RAND

Declaration

```
rand([N])
```

Description

The RAND([N]) function accepts zero or one parameter (N) and returns a random floating-point number that falls within the range of [0,1.0). N is called a random seed. If you want to obtain a random integer in the range of [I,j), you can use the FLOOR(I + RAND() * (j - i)) expression.

If you do not specify N, a random seed is generated during initialization. The RAND() function generates a random number based on this random seed. Therefore, the RAND() function generates a different random number each time the function is invoked.

If you specify N, N is used as the random seed to generate random numbers. The function generates random numbers based on whether N is a constant:

- If N is a constant, N is used as the random seed during initialization. Then, the RAND(N) function generates a random number based on the initialized value. If the values of N are the same, the generated random number sequences are the same.
- If N is a variable, such as a column name, N is used as the random seed to generate a random number each time the function is invoked. If the values of N are the same, the generated random numbers are the same.

In addition to SELECT statements, the RAND([N]) function can also be used in conjunction with the WHERE, ORDER BY, and GROUP BY clauses. Regardless of whether these clauses are used, the RAND([N]) function runs based on the preceding rules. For example, if you want to sort a table in a random way, execute the following SQL statement: select from t1 order by rand(). If you want to sample 100 rows of a table in a random way, execute the following SQL statement: select from t1 order by rand() limit 100.

Example

```
mysql> select a, b, rand() from t3;
+------+------+----------------------+
| a    | b    | rand()               |
+------+------+----------------------+
|    1 |    1 |   0.641815407799385 |
|    2 |    2 | 0.16825051248841966 |
|    3 |    3 |  0.9158063697775886 |
+------+------+----------------------+
3 rows in set (0.00 sec)

mysql> select a, b, rand() from t3;
+------+------+----------------------+
| a    | b    | rand()               |
+------+------+----------------------+
|    1 |    1 | 0.07428034215632857 |
|    2 |    2 |  0.6239826321825224 |
|    3 |    3 |   0.897072165177271 |
+------+------+----------------------+
3 rows in set (0.00 sec)

mysql> select a, b, rand(3) from t3;
+------+------+----------------------+
| a    | b    | rand(3)              |
+------+------+----------------------+
|    1 |    1 |  0.9057697559760601 |
|    2 |    2 | 0.37307905813034536 |
|    3 |    3 | 0.14808605345719125 |
+------+------+----------------------+
3 rows in set (0.00 sec)

mysql> select a, b, rand(3) from t3;
+------+------+----------------------+
| a    | b    | rand(3)              |
+------+------+----------------------+
|    1 |    1 |  0.9057697559760601 |
|    2 |    2 | 0.37307905813034536 |
|    3 |    3 | 0.14808605345719125 |
+------+------+----------------------+
3 rows in set (0.00 sec)

mysql> select a, b, rand(a), rand(b) from t3;
+------+------+----------------------+----------------------+
| a    | b    | rand(a)              | rand(b)              |
+------+------+----------------------+----------------------+
|    1 |    1 | 0.40540353712197724 | 0.40540353712197724 |
|    2 |    2 |  0.6555866465490187 |  0.6555866465490187 |
|    3 |    3 |  0.9057697559760601 |  0.9057697559760601 |
+------+------+----------------------+----------------------+
3 rows in set (0.00 sec)
```

Comparison functions

GREATEST

Declaration

```
GREATEST(value1, ...)
```

Description

This function returns the maximum value among the specified input values. This function performs the opposite operation of the LEAST() function.

You must specify a minimum of two arguments. The function generates an error if only one argument is specified. If one of the arguments is NULL, the function returns NULL.

If the specified arguments contain numeric values and strings, the system converts the strings to numeric values in an implicit way. If the conversion fails, the system reports an error.

Example

```
Oceanbase>select greatest(2,1), greatest('2',1,0), greatest('a','b','c'), greatest('a', NULL, 'c'), g
reatest('2014-05-15','2014-06-01')\G
    * 1. row *
                         greatest(2,1): 2
                     greatest('2',1,0): 2
                greatest('a','b','c'): c
             greatest('a', NULL, 'c'): NULL
    greatest('2014-05-15','2014-06-01'): 2014-06-01
    1 row in set (0.01 sec)
    Oceanbase>select greatest(2);
    ERROR 1582 (42000): Incorrect parameter count in the call to native function 'greatest'
```

LEAST

Declaration

```
LEAST(value1, ...)
```

Description

This function returns the minimum value among the specified arguments. This function performs the opposite operation of the GREATEST() function.

You must specify a minimum of two arguments. If one of the arguments is NULL, the function returns NULL.

If the specified arguments contain numeric values and strings, the system converts the strings to numeric values in an implicit way. If the conversion fails, the system reports an error.

Example

```
Oceanbase>select least(2, null), least('2',4,9), least('a','b','c'), least('a',NULL,'c'), least('2014
-05-15','2014-06-01')\G;
* 1. row *
                      least(2, null): NULL
                      least('2',4,9): 2
                  least('a','b','c'): a
                 least('a',NULL,'c'): NULL
least('2014-05-15','2014-06-01'): 2014-05-15
1 row in set (0.01 sec)
Oceanbase>select least(2);
ERROR 1582 (42000): Incorrect parameter count in the call to native function 'least'
```

ISNULL

Declaration

```
ISNULL(expr)
```

Description

This function checks whether a specified value of an expression is NULL. If the expr value is NULL, the ISNULL() function returns 1. Otherwise, the function returns 0.

The ISNULL() function can be used as an alternative to the equal sign (=) comparison. If the equal sign (=) operator is used to check whether an expression value is NULL, the returned result is invalid in most cases. The ISNULL() function provides some features that are the same as the IS NULL operator.

Example

```
Oceanbase>SELECT ISNULL(null), ISNULL('test'), ISNULL(123.456), ISNULL('10:00');
+--------------+----------------+-----------------+-----------------+
| ISNULL(null) | ISNULL('test') | ISNULL(123.456) | ISNULL('10:00') |
+--------------+----------------+-----------------+-----------------+
|            1 |              0 |               0 |               0 |
+--------------+----------------+-----------------+-----------------+
1 row in set (0.01 sec)

Oceanbase>SELECT ISNULL(null+1);
+-----------------+
| ISNULL(null+1) |
+-----------------+
|              1 |
+-----------------+
1 row in set (0.00 sec)
```

Process control functions

CASE

Declaration

```
CASE value WHEN [compare-value] THEN result [WHEN [compare-value] THEN result ...] [ELSE result] END
OR
CASE WHEN [condition] THEN result [WHEN [condition] THEN result ...] [ELSE result] END
```

Description

For the first case, the function returns the value that is equal to compare-value. For the second case, the function returns the first value whose preceding condition is true. If no matched results are available, the function returns the result of the ELSE part. If the ELSE part is unavailable, the function returns NULL.

Example

```
Oceanbase>select CASE 'b' when 'a' then 1 when 'b' then 2 END;
+-----------------------------------------------+
| CASE 'b' when 'a' then 1 when 'b' then 2 END |
+-----------------------------------------------+
|                                             2 |
+-----------------------------------------------+
1 row in set (0.01 sec)

Oceanbase>select CASE concat('a','b') when concat('ab','') then 'a' when 'b' then 'b' end;
+-----------------------------------------------------------------------------+
| CASE concat('a','b') when concat('ab','') then 'a' when 'b' then 'b' end |
+-----------------------------------------------------------------------------+
| a                                                                           |
+-----------------------------------------------------------------------------+
1 row in set (0.01 sec)

Oceanbase>select case when 1>0 then 'true' else 'false' end;
+--------------------------------------------+
| case when 1>0 then 'true' else 'false' end |
+--------------------------------------------+
| true                                       |
+--------------------------------------------+
1 row in set (0.00 sec)
```

IF

Declaration

```
IF(expr1,expr2,expr3)
```

Description

This function returns the expr2 value if the expr1 value is TRUE. The expr1 value is TRUE if expr1 is a non-NULL value that is not equal to 0. Otherwise, the function returns the expr3 value.

The IF() function returns numeric values or strings based on the data types of arguments.

If one of the expr2 and expr3 values is NULL, the data type of the IF() function result is the same as that of the non-NULL expression value.

Example

```
Oceanbase>select if(5>6, 'T','F'), if (5>6, 1, 0), if(null, 'True', 'False'), if(0, 'True', 'False')\
G
* 1. row *
         if(5>6, 'T','F'): F
           if (5>6, 1, 0): 0
if(null, 'True', 'False'): False
   if(0, 'True', 'False'): False
1 row in set (0.01 sec)
```

IFNULL

Declaration

```
IFNULL(expr1, expr2)
```

Description

This function returns expr1 if expr1 is not NULL. Otherwise, the function returns expr2. The IFNULL() function returns numeric values or strings based on the data types of arguments.

The following table describes the rules used by the IF() function to return data of the default type.

Expression	Returned value
expr1 or expr2 returns a string.	String
expr2 or expr3 returns a floating-point number.	Float-pointing number
expr2 or expr3 returns an integer.	Integer

If the expr2 and expr3 values are strings and one of the strings is case-sensitive, the returned value is case-sensitive.

Example

```
Oceanbase>SELECT IFNULL('abc', null), IFNULL(NULL+1, NULL+2), IFNULL(1/0, 0/1);
+---------------------+-----------------------+------------------+
| IFNULL('abc', null) | IFNULL(NULL+1, NULL+2) | IFNULL(1/0, 0/1) |
+---------------------+-----------------------+------------------+
| abc                 |                  NULL |           0.0000 |
+---------------------+-----------------------+------------------+
1 row in set (0.01 sec)
```

NULLIF

Declaration

```
NULLIF(expr1, expr2)
```

Description

If the expr1 value is equal to the expr2 value, the function returns NULL. Otherwise, the function returns the expr1 value. This function performs the same operation as the CASE WHEN expr1 = expr2 THEN NULL ELSE expr1 END statement.

This function evaluates the expr1 value twice if the expr1 and expr2 values are not equal.

Example

```
Oceanbase>SELECT NULLIF('ABC', 123), NULLIF('123',123), NULLIF(NULL, 'abc');
+--------------------+--------------------+---------------------+
| NULLIF('ABC', 123) | NULLIF('123',123) | NULLIF(NULL, 'abc') |
+--------------------+--------------------+---------------------+
| ABC                | NULL               | NULL                |
+--------------------+--------------------+---------------------+
1 row in set, 1 warning (0.01 sec)
```

ORA_DECODE

Declaration

```
 ora_decode (condition, value 1, return value 1, value 2, return value 2, ... value n, return value n, default value)
```

Description

The ORA_DECODE() function is equivalent to the DECODE() function provided by Oracle. ApsaraDB for OceanBase 1.0 is compatible with MySQL, and also supports some Oracle functions. The names of the supported Oracle functions are prefixed with ORA_ in ApsaraDB for OceanBase 1.0.

The following section describes the meaning of this function:

```
IF condition = value 1
THEN RETURN (return value 1)
ELSIF condition = value 2
THEN RETURN (return value 2)
......
ELSIF condition = value n
THEN RETURN (return value n)
ELSE RETURN (default value)
END IF
```

17.1.4.3.2. Aggregate functions

Aggregate functions perform calculations on a set of values and return a single value. Aggregate functions ignore NULL values. In most cases, aggregate functions are used in conjunction with GROUP BY clauses in SELECT statements.

All aggregate functions are deterministic. Each time you call aggregate functions by using the same set of input values, the aggregate functions return the same value.

In ApsaraDB for OceanBase, you can specify only one expression value for each aggregate function. For example, COUNT(c1, c2) is not supported, and only COUNT(c1) is supported.

AVG

Declaration

```
AVG(([DISTINCT] expr)
```

Description

This function returns the average value of a specified data set. The function ignores the NULL values in the specified data set. The DISTINCT keyword is used to return the average value of the distinct expr values. If no matched rows are found, the AVG() function returns NULL.

Example

```
Oceanbase>select * from oceanbasetest;
+----+------+------+
| id | ip   | ip2  |
+----+------+------+
|  1 |    4 | NULL |
|  3 |    3 | NULL |
|  4 |    3 | NULL |
+----+------+------+
3 rows in set (0.01 sec)

Oceanbase>select avg(ip2), avg(ip), avg(distinct(ip)) from oceanbasetest;
+----------+---------+-------------------+
| avg(ip2) | avg(ip) | avg(distinct(ip)) |
+----------+---------+-------------------+
|     NULL |  3.3333 |            3.5000 |
+----------+---------+-------------------+
1 row in set (0.00 sec)

Oceanbase>select avg(distinct(ip)),avg(ip),avg(ip2) from oceanbasetest;
+-------------------+---------+----------+
| avg(distinct(ip)) | avg(ip) | avg(ip2) |
+-------------------+---------+----------+
|            3.5000 |  3.3333 |     NULL |
+-------------------+---------+----------+
1 row in set (0.00 sec)
```

COUNT

Declaration

```
COUNT([DISTINCT] expr)
```

Description

This function returns the number of non-NULL values in the rows retrieved by the SELECT statement. If no matched rows are found, COUNT() returns 0. The DISTINCT keyword is used to return the number of the distinct expr values.

COUNT(*) returns the number of retrieved rows, regardless of whether they contain NULL values.

Example

```
Oceanbase>select * from oceanbasetest;
+----+------+------+
| id | ip   | ip2  |
+----+------+------+
|  1 |    4 | NULL |
|  3 |    3 | NULL |
|  4 |    3 | NULL |
+----+------+------+
3 rows in set (0.00 sec)

Oceanbase>select count(ip2), count(ip), count(distinct(ip)), count(*) from oceanbasetest;
+------------+-----------+--------------------+----------+
| count(ip2) | count(ip) | count(distinct(ip)) | count(*) |
+------------+-----------+--------------------+----------+
|          0 |         3 |                  2 |        3 |
+------------+-----------+--------------------+----------+
1 row in set (0.00 sec)
```

MAX

Declaration

```
MAX([DISTINCT] expr)
```

Description

This function returns the maximum value of a specified data set.

You can specify strings in the MAX() function. In this case, the maximum string is returned. The DISTINCT keyword can be used to return the maximum value of the distinct expr values. MAX() returns the same maximum value regardless of whether you omit the DISTINCT keyword.

Assume that table a has three rows. For the id column, the values are 1, 2, and 3. For the num column, the values are 10, 20, and 30.

Example

```
Oceanbase>SELECT MAX(num) FROM a;
+-----------------+
| MAX(num)        |
+-----------------+
|              30 |
+-----------------+
1 row in set (0.00 sec)
```

MIN

Declaration

```
MIN([DISTINCT] expr)
```

Description

This function returns the minimum value of a specified data set.

You can specify strings in the MIN() function. In this case, the minimum string is returned. The DISTINCT keyword can be used to return the minimum value of the distinct expr values. MIN() returns the same minimum value regardless of whether you omit the DISTINCT keyword.

Assume that table a has three rows. For the id column, the values are 1, 2, and 3. For the num column, the values are 10, 20, and 30.

Example

```
Oceanbase>SELECT MIN(num) FROM a;
+-----------------+
| MIN(num)        |
+-----------------+
|              10 |
+-----------------+
1 row in set (0.00 sec)
```

SUM

Declaration

```
SUM([DISTINCT] expr)
```

Description

This function returns the sum value of the expr values. If no rows are found in expr, the SUM() function returns NULL. You can use the DISTINCT keyword to return the sum value of the distinct expr values.

If no matched rows are found, the SUM() function returns NULL.

Example

```
Oceanbase>select * from oceanbasetest;
+------+------+------+
| id   | ip   | ip2  |
+------+------+------+
|    1 |    4 | NULL |
|    3 |    3 | NULL |
|    4 |    3 | NULL |
+------+------+------+
3 rows in set (0.00 sec)

Oceanbase>select sum(ip2),sum(ip),sum(distinct(ip)) from oceanbasetest;
+----------+---------+-------------------+
| sum(ip2) | sum(ip) | sum(distinct(ip)) |
+----------+---------+-------------------+
|     NULL |      10 |                 7 |
+----------+---------+-------------------+
1 row in set (0.00 sec)
```

GROUP_CONCAT

Declaration

```
GROUP_CONCAT([DISTINCT] expr)
```

Description

This function concatenates non-NULL strings from a specified group into a single string.

```
GROUP_CONCAT([DISTINCT] expr [,expr ...]
    [ORDER BY {unsigned_integer | col_name | expr}
     ASC | DESC]
    [SEPARATOR str_val])
```

Example

```
Oceanbase> select * from book; //The table named book (book number, book title, publisher)
+--------+--------------------------------+-----------------------------+
| bookid | bookname                       | publishname                 |
+--------+--------------------------------+-----------------------------+
|      1 | git help                       | alibaba group publisher     |
|      2 | MySQL Optimization                  | Zhejiang University Press          |
|      3 | Java Programming Guide                   | Machinery Industry Press              |
|      3 | Java Programming Guide                      | Machinery Industry Press   |
|     4 | Large-Scale Distributed Storage System           | Machinery Industry Press              |
+--------+--------------------------------+-----------------------------+
5 rows in set (0.00 sec)

//Retrieve book titles.
Oceanbase>select group_concat(bookname) from book group by bookname;
+-----------------------------------+
| group_concat(bookname)            |
+-----------------------------------+
| git help                          |
| Java Programming Guide, Java Programming Guide         |
| MySQL Optimization                     |
| Large-Scale Distributed Storage System              |
+-----------------------------------+
4 rows in set (0.00 sec)

//Retrieve distinct book titles.
Oceanbase>select group_concat(distinct(bookname)) from book group by bookname;
+-----------------------------------+
| group_concat(distinct(bookname)) |
+-----------------------------------+
| git help                          |
| Java Programming Guide                     |
| MySQL Optimization                    |
| Large-Scale Distributed Storage System             |
+-----------------------------------+
4 rows in set (0.01 sec)

//Query book titles and publisher information, group the books by book title, and sort the publisher
names in reverse alphabetical order.
Oceanbase>select bookname, group_concat(publishname order by publishname desc separator  ';' ) from b
ook group by bookname;
+--------------------------------+-------------------------------------------------------------------------
--+
| bookname                       | group_concat(publishname order by publishname desc separator  ';'
) |
+--------------------------------+-------------------------------------------------------------------------
--+
| git help                       | alibaba group publisher
|
| Java Programming Guide                   | Mechanical Industry Press, Mechanical Industry Press
|
| MySQL Optimization                  | Zhejiang University Press
|
| Large-Scale Distributed Storage System           | Machinery Industry Press
|
+--------------------------------+-------------------------------------------------------------------------
--+
4 rows in set (0.00 sec)
```

17.1.4.3.3. Analytic functions

Overview

Analytic functions, also called **window functions** in some database services, are similar to aggregate functions. The similarity lies in that the calculations are performed based on a set of rows. The difference is that an aggregate function returns only one row for each set of rows, but an analytic function returns one or more rows for each set of rows. Each of the rows returned by the analytic function is the result of window-based logic computing. Analytic functions can optimize queries that contain self joins in a significant way.

Features

ApsaraDB for OceanBase supports the following analytic functions:

- SUM
- MIN
- MAX
- COUNT
- AVG
- GROUP_CONCAT, a function that is specific to ApsaraDB for OceanBase
- ROW_NUMBER
- RANK
- DENSE_RANK
- PERCENT_RANK
- CUME_DIST
- FIRST_VALUE
- LAST_VALUE
- NTH_VALUE
- NTILE
- LEAD
- LAG
- RATIO_TO_REPORT
- STDDEV
- VARIANCE
- STDDEV_SAMP
- STDDEV_POP
- LISTAGG

ApsaraDB for OceanBase does not support the following window functions that are supported by Oracle:

- CORR
- COVAR_POP
- COVAR_SAMP
- MEDIAN
- PERCENTILE_CONT
- PERCENTILE_DISC

- VAR_POP
- VAR_SAMP
- REGR_(Linear Regression) Functions

Syntax of analytic functions

A window is also called a frame. ApsaraDB for OceanBase supports the ROWS and RANGE options for the FRAME clauses. The ROWS option defines frames based on the offsets that are the differences in physical row numbers from the current row number. The RANGE option defines frames based on the offsets that are the differences in logical row values from the current row value.

Syntax of analytic_function: analytic_function([arguments]) **OVER** (analytic_clause)

Syntax of analytic_clause: [query_partition_clause] [order_by_clause [windowing_clause]]

Syntax of query_partition_clause: **PARTITION BY** { expr[, expr]... | (expr[, expr]...) }

Syntax of order_by_clause: **ORDER** [**SIBLINGS**] **BY**{ expr | position | c_alias } [**ASC** | **DESC**] [**NULLS FIRST** | **NULLS LAST**] [, { expr | position | c_alias } [**ASC** | **DESC**][**NULLS FIRST** | **NULLS LAST**]]...

Syntax of windowing_clause

Syntax of windowing_clause:

{ **ROWS** | **RANGE** } { **BETWEEN** { **UNBOUNDED PRECEDING** | **CURRENT ROW** | value_expr { **PRECEDING** | **FOLLOWING** } } **AND**{ **UNBOUNDED FOLLOWING** | **CURRENT ROW** | value_expr { **PRECEDING** | **FOLLOWING** } } | { **UNBOUNDED PRECEDING** | **CURRENT ROW**| value_expr **PRECEDING**}}

The syntax of windowing_clause is compatible with that of Oracle.

Description

The following content describes the syntax and features of analytic functions, and also compares analytic functions used in ApsaraDB for OceanBase and Oracle.

sum/min/max/count/avg

Syntax of the AVG function: **AVG**([**DISTINCT** | **ALL**] expr) [**OVER**(analytic_clause)]

Syntax of the SUM function: **SUM**([**DISTINCT** | **ALL**] expr) [**OVER** (analytic_clause)]

Syntax of the MIN function: **MIN**([**DISTINCT** | **ALL**] expr) [**OVER** (analytic_clause)]

Syntax of the MAX function: **MAX**([**DISTINCT** | **ALL**] expr) [**OVER** (analytic_clause)]

Syntax of the COUNT function: **COUNT** ({ * | [**DISTINCT** | **ALL**] expr }) [**OVER** (analytic_clause)]

The preceding analytic functions have corresponding aggregate functions. The SUM function returns the sum of the expr values. The MIN function returns the minimum expr value. The MAX function returns the maximum expr value. The COUNT function returns the number of rows queried for a window. The AVG function returns the average expr value.

The COUNT function does not return null. If you specify expr, the function returns the number of non-null expr values. If you use an asterisk (*), the COUNT(*) function returns the number of all the rows.

```
create table exployees(last_name char(10), salary decimal, job_id char(32));
insert into exployees values('jim', 2000, 'cleaner');
insert into exployees values('mike', 12000, 'engineering');
insert into exployees values('lily', 13000, 'engineering');
insert into exployees values('tom', 11000, 'engineering');
OceanBase(TEST@TEST)>select last_name, sum(salary) over(partition by job_id) totol_s, min(salary) ove
r(partition by job_id) min_s, max(salary) over(partition by job_id) max_s, count(*) over(partition by
job_id) count_s from exployees;
+------------+---------+-------+-------+---------+
| LAST_NAME  | TOTOL_S | MIN_S | MAX_S | COUNT_S |
+------------+---------+-------+-------+---------+
| jim        |    2000 |  2000 |  2000 |       1 |
| mike       |   36000 | 11000 | 13000 |       3 |
| lily       |   36000 | 11000 | 13000 |       3 |
| tom        |   36000 | 11000 | 13000 |       3 |
+------------+---------+-------+-------+---------+
```

The syntax of the preceding five functions in ApsaraDB for OceanBase is not fully compatible with that in Oracle.

If you use the DISTINCT keyword in the SUM, COUNT, or AVG function, you can specify only query_partition_clause in analytic_clause. order_by_clause and windowing_clause are not allowed. **ApsaraDB for OceanBase does not support this option. If you use DISTINCT in ApsaraDB for OceanBase, you can also specify order_by_clause and windowing_clause.**

ApsaraDB for OceanBase does not support NULLS FIRST or NULLS LAST in order_by_clause.

For more information about the support for analytic_clause, see the "Syntax of windowing_clause" section.

nth_value/first_value/last_value

Syntax of the NTH_VALUE function: **NTH_VALUE** (measure_expr, n) [**FROM** { **FIRST** | **LAST** }][{ **RESPECT** | **IGNORE** } **NULLS**] **OVER** (analytic_clause)

The NTH_VALUE function, as its name suggests, returns the Nth value in the window defined by analytic_clause. [**FROM** { **FIRST** | **LAST** }] specifies whether to begin the calculation from the first or last row of the window. By default, FROM FIRST is used. [{RESPECT | IGNORE}] NULLS specifies whether to ignore null during the calculation. The N value must be a positive integer. If the N value is null, the function returns an error. If the N value is greater than the number of all the rows in the window, the function returns null.

Syntax of the FIRST_VALUE or LAST_VALUE function: **FIRST_VALUE** or LAST_VALUE { (expr) [{**RESPECT** | **IGNORE**} NULLS] | (expr [{**RESPECT** | **IGNORE**} NULLS])} **OVER** (analytic_clause)

The FIRST_VALUE function, as its name suggests, returns the first value in a set of values. The LAST_VALUE function returns the last value in a set of values.

```
create table exployees(last_name char(10), salary decimal, job_id char(32));
insert into exployees values('jim', 2000, 'cleaner');
insert into exployees values('mike', 12000, 'engineering');
insert into exployees values('lily', 13000, 'engineering');
insert into exployees values('tom', 11000, 'engineering');
OceanBase(TEST@TEST)>select last_name, first_value(salary) over(partition by job_id) totol_s, last_va
lue(salary) over(partition by job_id) min_s, max(salary) over(partition by job_id) max_s from exploye
es;
+------------+---------+-------+-------+
| LAST_NAME  | TOTOL_S | MIN_S | MAX_S |
+------------+---------+-------+-------+
| jim        |    2000 |  2000 |  2000 |
| mike       |   12000 | 11000 | 13000 |
| lily       |   12000 | 11000 | 13000 |
| tom        |   12000 | 11000 | 13000 |
+------------+---------+-------+-------+
```

The syntax of the preceding three functions is compatible with that of Oracle. No corresponding aggregate functions are available for the three functions.

lead/lag

The LAG or LEAD function allows you to query the value of the same field from a row that appears before or after the current row at the N offset. Self joins can also perform these operations. However, the LAG and LEAD analytic functions enable more efficient queries than self joins. The following syntax is used:

lag/lead { (value_expr [, offset [, default]]) [{ **RESPECT** | **IGNORE** } **NULLS**] | (value_expr [{ **RESPECT** | **IGNORE** } **NULLS**] [, offset [, default]])} **OVER** ([query_partition_clause] order_by_clause)

value_expr specifies the field whose values are to be compared, and offset specifies the offset from the current row that is defined by value_expr. The default value of the default parameter is null. This indicates that if you do not specify the default parameter for the LAG or LEAD function, a null value is returned. Assume that you specify parameters for the LAG function and the current row is the fourth row in the table. If you set the offset parameter to 6, the serial number of the row that you want to query is -2. This goes beyond the scope of the table. In this case, the function returns the default value.

[{ **RESPECT** | **IGNORE** } **NULLS**] specifies whether to ignore null during the calculation. The default value is **RESPECT**. This indicates that the null value is not ignored.

Take note of this point: The LEAD or LAG function must be followed by order_by_clause because this clause specifies the order in which the values in a column are sorted. The function can calculate the positions of the current row and the previous or subsequent rows only after the values are sorted. query_partition_clause is optional. If you do not specify this clause, the operation of the function applies to the global data.

```
create table exployees(last_name char(10), salary decimal, job_id char(32));
insert into exployees values('jim', 2000, 'cleaner');
insert into exployees values('mike', 12000, 'engineering');
insert into exployees values('lily', 13000, 'engineering');
insert into exployees values('tom', 11000, 'engineering');
OceanBase(TEST@TEST)>select last_name, lead(salary) over(order by salary) lead, lag(salary) over(orde
r by salary) lag from exployees;
+------------+-------+-------+
| LAST_NAME  | LEAD  | LAG   |
+------------+-------+-------+
| jim        | 11000 |  NULL |
| tom        | 12000 |  2000 |
| mike       | 13000 | 11000 |
| lily       |  NULL | 12000 |
+------------+-------+-------+
```

The LAG and LEAD functions in ApsaraDB for OceanBase are compatible with those in Oracle. Take note of the following point: In Oracle and ApsaraDB for OceanBase, you can specify whether to ignore null in the results returned by the LEAD or LAG function. No corresponding aggregate function is available for the LEAD or LAG function.

stddev/variance/stddev_samp/stddev_pop

Syntax of the VARIANCE function: **VARIANCE**([**DISTINCT** | **ALL**] expr) [**OVER** (analytic_clause)]. The VARIANCE function returns the variance of the expr value. You can set expr to a value of a numeric data type or non-numeric data type that can be converted to a numeric data type. The function returns the variance of the same data type as the argument.

Syntax of the STDDEV function: **STDDEV**([**DISTINCT** | **ALL**] expr) [**OVER** (analytic_clause)]. The STDDEV function returns the standard deviation of the expr value. The STDDEV and VARIANCE functions have the same syntax and parameter types.

Syntax of the STDDEV_SAMP function: **STDDEV_SAMP**(expr) [**OVER** (analytic_clause)]

Syntax of the STDDEV_POP function: **STDDEV_POP**(expr) [**OVER** (analytic_clause)]

```
create table exployees(last_name char(10), salary decimal, job_id char(32));
insert into exployees values('jim', 2000, 'cleaner');
insert into exployees values('mike', 12000, 'engineering');
insert into exployees values('lily', 13000, 'engineering');
insert into exployees values('tom', 11000, 'engineering');
OceanBase(TEST@TEST)>select last_name, stddev(salary) over(order by salary) std, variance(salary) ove
r(order by salary) var, stddev_pop(salary) over() std_pop, stddev_samp(salary) over() from exployees;
+------------+----------------------------------------+-----------------------------------------+-
----------------------------------------+-----------------------------------------+
| LAST_NAME  | STD                                    | VAR                                     |
STD_POP                                | STDDEV_SAMP(SALARY)OVER()               |
+------------+----------------------------------------+-----------------------------------------+-
----------------------------------------+-----------------------------------------+
| jim        |                                      0 |                                       0 |
4387.48219369606103020319415370815478043 8 | 5066.228051190221210556248068679881679227 |
| tom        | 6363.961030678927719607599258943641353564 |                                40500000 |
4387.48219369606103020319415370815478043 8 | 5066.228051190221210556248068679881679227 |
| mike       | 5507.570547286102020605808770872073470916 | 30333333.333333333333333333333333333333 |
4387.48219369606103020319415370815478043 8 | 5066.228051190221210556248068679881679227 |
| lily       | 5066.228051190221210556248068679881679227 | 25666666.666666666666666666666666666667 |
4387.48219369606103020319415370815478043 8 | 5066.228051190221210556248068679881679227 |
+------------+----------------------------------------+-----------------------------------------+-
----------------------------------------+-----------------------------------------+
```

The syntax of the preceding four functions in ApsaraDB for OceanBase is not fully compatible with that in Oracle.

If you use the DISTINCT keyword in the VARIANCE or STDDEV function, you can specify only query_partition_clause in analytic_clause. order_by_clause and windowing_clause are not allowed. **ApsaraDB for OceanBase does not support this option. If you use DISTINCT in ApsaraDB for OceanBase, you can also specify order_by_clause and windowing_clause.** For the STDDEV_POP and STDDEV_SAMP functions, you cannot use the DISTINCT or ALL keyword. **ApsaraDB for OceanBase does not support this option.**

ApsaraDB for OceanBase does not support NULLS FIRST or NULLS LAST in order_by_clause.

For more information about the support for analytic_clause, see the "Syntax of windowing_clause" section.

ntile

Syntax of the NTILE function: **NTILE**(expr) **OVER** ([query_partition_clause] order_by_clause). The NTILE function divides sorted partition rows into a specified number of groups, and the size of each group is the same if possible. The function assigns each row to a group by giving each row a group number. If the expr value is null, the function returns null.

```
create table exployees(last_name char(10), salary decimal, job_id char(32));
insert into exployees values('jim', 2000, 'cleaner');
insert into exployees values('mike', 12000, 'engineering');
insert into exployees values('lily', 13000, 'engineering');
insert into exployees values('tom', 11000, 'engineering');
OceanBase(TEST@TEST)>select last_name, ntile(10) over(partition by job_id order by salary) ntl from e
xployees;
+------------+------+
| LAST_NAME  | NTL  |
+------------+------+
| jim        |    1 |
| tom        |    1 |
| mike       |    2 |
| lily       |    3 |
+------------+------+
```

The syntax of the NTILE function in ApsaraDB for OceanBase is compatible with that in Oracle. Take note of this point: You must specify order_by_clause for the NTILE analytic function. No corresponding aggregate function is available for the NTILE function.

row_number

Syntax of the ROW_NUMBER function: **ROW_NUMBER( ) OVER** ([query_partition_clause] order_by_clause). query_partition_clause is optional. The ROW_NUMBER function can run as expected only after the values are sorted. This function assigns a unique serial number to each row after the rows are sorted by applying the ORDER BY clause on expr values.

```
create table exployees(last_name char(10), salary decimal, job_id char(32));
insert into exployees values('jim', 2000, 'cleaner');
insert into exployees values('mike', 12000, 'engineering');
insert into exployees values('lily', 13000, 'engineering');
insert into exployees values('tom', 11000, 'engineering');
OceanBase(TEST@TEST)>select last_name, row_number() over(partition by job_id order by salary) ntl fro
m exployees;
+------------+------+
| LAST_NAME  | NTL  |
+------------+------+
| jim        |    1 |
| tom        |    1 |
| mike       |    2 |
| lily       |    3 |
+------------+------+
```

The syntax of the ROW_NUMBER function in ApsaraDB for OceanBase is compatible with that in Oracle. Take note of this point: You must specify order_by_clause for the ROW_NUMBER analytic function. No corresponding aggregate function is available for the ROW_NUMBER function.

rank/dense_rank/percent_rank

Syntax of the RANK function: **RANK( ) OVER** ([query_partition_clause] order_by_clause). The RANK function is an analytic function and ranks all the rows for a given column that is specified by order_by_clause. In the following example, all the employees are ranked by salary:

Syntax of the DENSE_RANK function: **DENSE_RANK( ) OVER**([query_partition_clause] order_by_clause). The DENSE_RANK function acts like the RANK function except that DENSE_RANK assigns consecutive ranks.

Syntax of the PERCENT_RANK function: **PERCENT_RANK( ) OVER** ([query_partition_clause] order_by_clause). The PERCENT_RANK function acts like the RANK function except that PERCENT_RANK calculates the rank of a given row as a percentage.

```
create table exployees(last_name char(10), salary decimal, job_id char(32));
insert into exployees values('jim', 2000, 'cleaner');
insert into exployees values('mike', 12000, 'engineering');
insert into exployees values('lily', 13000, 'engineering');
insert into exployees values('tom', 11000, 'engineering');
OceanBase(TEST@TEST)>select last_name, rank() over(partition by job_id order by salary) rank, dense_r
ank() over(partition by job_id order by salary) dense_rank, percent_rank() over(partition by job_id o
rder by salary) percent_rank from exployees;
+------------+------+------------+--------------+
| LAST_NAME  | RANK | DENSE_RANK | PERCENT_RANK |
+------------+------+------------+--------------+
| jim        |    1 |          1 |            0 |
| tom        |    1 |          1 |            0 |
| mike       |    2 |          2 |           .5 |
| lily       |    3 |          3 |            1 |
+------------+------+------------+--------------+
```

In the preceding three functions, the syntax of the RANK function in ApsaraDB for OceanBase is compatible with that in Oracle. Take note of this point: You must specify order_by_clause for the RANK function. Take note of this point: The syntax of the RANK aggregate function differs from that of the RANK analytic function. For more information, see the syntax of the RANK function in the "Aggregate functions" section. The syntax of the DENSE_RANK function in ApsaraDB for OceanBase is compatible with that in Oracle. Take note of this point: You must specify order_by_clause for the DENSE_RANK function. Take note of this point: The syntax of the DENSE_RANK aggregate function differs from that of the DENSE_RANK analytic function. For more information, see the syntax of the DENSE_RANK function in the "Aggregate functions" section. The syntax of the PERCENT_RANK function in ApsaraDB for OceanBase is compatible with that in Oracle. Take note of this point: You must specify order_by_clause for the PERCENT_RANK function. Take note of this point: The syntax of the PERCENT_RANK aggregate function differs from that of the PERCENT_RANK analytic function. For more information, see the syntax of the PERCENT_RANK function in the "Aggregate functions" section.

cume_dist

Syntax of the CUME_DIST function: **CUME_DIST** () **OVER** ([query_partition_clause] order_by_clause). The CUME_DIST function calculates the cumulative distribution of a value. CUME_DIST returns a value that is greater than 0 and less than or equal to 1. Assume that the values in a specified column are sorted in ascending order. The CUME_DIST analytic function calculates the percentage of each value that is less than or equal to the current value for the specified column. The following example groups the employees by job_id and ranks the salary percentile of each employee in a specified window.

```
create table exployees(last_name char(10), salary decimal, job_id char(32));
insert into exployees values('jim', 2000, 'cleaner');
insert into exployees values('mike', 12000, 'engineering');
insert into exployees values('lily', 13000, 'engineering');
insert into exployees values('tom', 11000, 'engineering');
OceanBase(TEST@TEST)>select last_name, cume_dist() over(partition by job_id order by salary) rank fro
m exployees;
+------------+-----------------------------------------+
| LAST_NAME  | RANK                                    |
+------------+-----------------------------------------+
| jim        |                                       1 |
| tom        | .33333333333333333333333333333333333333 |
| mike       | .66666666666666666666666666666666666667 |
| lily       |                                       1 |
+------------+-----------------------------------------+
```

The syntax of the CUME_DIST function in ApsaraDB for OceanBase is compatible with that in Oracle. Take note of this point: The syntax of the CUME_DIST function includes query_partition_clause and order_by_clause. Take note of this point: The syntax of the CUME_DIST aggregate function differs from that of the CUME_DIST analytic function. For more information, see the syntax of the CUME_DIST function in the "Aggregate functions" section.

ratio_to_report

Syntax of the RATIO_TO_REPORT function: **RATIO_TO_REPORT** (expr) **OVER** ([query_partition_clause]). The RATIO_TO_REPORT function calculates the expr value and the ratio of the value to the sum of a set of values. If the expr value is null, the RATIO_TO_REPORT function returns null.

The sum of a set of values is determined by query_partition_clause. If you omit this clause, RATIO_TO_REPORT calculates the ratio of a value to the sum of all the expr values. You cannot use RATIO_TO_REPORT or other analytic functions in expr. However, you can use built-in functions in expr. This is similar to the expr values in aggregate functions. The following example is used to describe how to use RATIO_TO_REPORT to calculate the ratio of each employee's salary to the total salary of all the employees in the company:

```
create table exployees(last_name char(10), salary decimal, job_id char(32));
insert into exployees values('jim', 2000, 'cleaner');
insert into exployees values('mike', 12000, 'engineering');
insert into exployees values('lily', 13000, 'engineering');
insert into exployees values('tom', 11000, 'engineering');
OceanBase(TEST@TEST)>select last_name, salary, ratio_to_report(salary) over() as salary_ratio from ex
ployees where job_id = 'engineering';
+------------+--------+-----------------------------------------+
| LAST_NAME  | SALARY | SALARY_RATIO                            |
+------------+--------+-----------------------------------------+
| mike       |  12000 | .33333333333333333333333333333333333333 |
| lily       |  13000 | .36111111111111111111111111111111111111 |
| tom        |  11000 | .30555555555555555555555555555555555556 |
+------------+--------+-----------------------------------------+
```

The syntax of the RATIO_TO_REPORT function in ApsaraDB for OceanBase is compatible with that in Oracle. Take note of this point: You cannot specify order_by_clause for the RATIO_TO_REPORT analytic function.

Execution plans

The WINDOW FUNCTION operator is used to implement analytic functions in ApsaraDB for OceanBase. WINDOW FUNCTION works based on the partitioned and sorted output of the lower-level operator. Therefore, the system assigns a SORT operator if the SORT operator is required.

Example

```
mysql> explain select row_number() over (partition by table_id order by partition_id) from __all_root
_table;
+-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-----------------------------------+
| Query Plan
|
+-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-----------------------------------+
| ==================================================
|ID|OPERATOR       |NAME            |EST. ROWS|COST|
--------------------------------------------------
|0 |WINDOW FUNCTION|                |1000     |2240|
|1 | SORT          |                |1000     |2049|
|2 |  TABLE SCAN   |__all_root_table|1000     |499 |
==================================================

Outputs & filters:
-------------------------------------
  0 - output([T_WIN_FUN_ROW_NUMBER()]), filter(nil),
      win_expr(T_WIN_FUN_ROW_NUMBER()), partition_by([__all_root_table.table_id]), order_by([__all_ro
ot_table.partition_id, ASC]), window_type(RANGE), upper(UNBOUNDED PRECEDING), lower(UNBOUNDED FOLLOWI
NG)
  1 - output([__all_root_table.table_id], [__all_root_table.partition_id]), filter(nil), sort_keys([_
_all_root_table.table_id, ASC], [__all_root_table.partition_id, ASC])
  2 - output([__all_root_table.table_id], [__all_root_table.partition_id]), filter(nil),
      access([__all_root_table.table_id], [__all_root_table.partition_id]), partitions(p0)
```

Remove a SORT operator

If the optimizer recognizes that the output of the lower-level operator is sorted, the optimizer does not assign a SORT operator. In this case, (tenant_id, table_id) is the prefix of the primary key in the __all_root_table table.

```
explain select row_number() over (partition by tenant_id, table_id order by partition_id) from __all_
root_table;
+-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
--+
| Query Plan
|
+-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
--+
| ==================================================
|ID|OPERATOR       |NAME            |EST. ROWS|COST|
--------------------------------------------------
|0 |WINDOW FUNCTION|                |1000     |690 |
|1 | TABLE SCAN    |__all_root_table|1000     |499 |
==================================================

Outputs & filters:
-------------------------------------
  0 - output([T_WIN_FUN_ROW_NUMBER()]), filter(nil),
      win_expr(T_WIN_FUN_ROW_NUMBER()), partition_by([__all_root_table.tenant_id], [__all_root_table.
table_id]), order_by([__all_root_table.partition_id, ASC]), window_type(RANGE), upper(UNBOUNDED PRECE
DING), lower(UNBOUNDED FOLLOWING)
  1 - output([__all_root_table.tenant_id], [__all_root_table.table_id], [__all_root_table.partition_i
d]), filter(nil),
      access([__all_root_table.tenant_id], [__all_root_table.table_id], [__all_root_table.partition_i
d]), partitions(p0)
 |
```

Combine calculations

When multiple analytic functions use the same sorted subset for calculation, all the functions use the same operator instead of multiple independent operators. This reduces memory overhead and range checks in an effective way.

In the following example, the calculation of the two analytic functions ROW_NUMBER() and RANK() can use the same sorted output. Therefore, only one SORT operator is assigned.

```
mysql> explain select row_number() over (partition by table_id order by partition_id), rank() over (p
artition by table_id, partition_id order by svr_ip) from __all_root_table;
| ====================================================
|ID|OPERATOR       |NAME            |EST. ROWS|COST|
----------------------------------------------------
|0 |WINDOW FUNCTION|                |1000     |3017|
|1 | SORT          |                |1000     |2826|
|2 |  TABLE SCAN   |__all_root_table|1000     |499 |
====================================================

Outputs & filters:
-------------------------------------
  0 - output([T_WIN_FUN_ROW_NUMBER()], [T_WIN_FUN_RANK()]), filter(nil),
      win_expr(T_WIN_FUN_RANK()), partition_by([__all_root_table.table_id], [__all_root_table.partiti
on_id]), order_by([__all_root_table.svr_ip, ASC]), window_type(RANGE), uppe                   r(UNBOU
NDED PRECEDING), lower(UNBOUNDED FOLLOWING)
      win_expr(T_WIN_FUN_ROW_NUMBER()), partition_by([__all_root_table.table_id]), order_by([__all_ro
ot_table.partition_id, ASC]), window_type(RANGE), upper(UNBOUNDED PRECEDING                   ), lowe
r(UNBOUNDED FOLLOWING)
  1 - output([__all_root_table.table_id], [__all_root_table.partition_id], [__all_root_table.svr_ip])
, filter(nil), sort_keys([__all_root_table.table_id, ASC], [__all_root_tabl                   e.parti
tion_id, ASC], [__all_root_table.svr_ip, ASC])
  2 - output([__all_root_table.table_id], [__all_root_table.partition_id], [__all_root_table.svr_ip])
, filter(nil),
      access([__all_root_table.table_id], [__all_root_table.partition_id], [__all_root_table.svr_ip])
, partitions(p0)
```

Store data on disks

Analytic functions also support storing intermediate data on disks during execution. This implements the same logic as the SORT operator.

17.1.4.3.4. Information functions

FOUND_ROWS

Declaration

```
found_rows()
```

Description

You may use a LIMIT clause in a SELECT statement to limit the number of rows that are returned from the database server to the client. In some cases, you need to retrieve the actual number of rows that the statement returns if the statement does not have the LIMIT clause. If you do not want to execute the statement again, you can use SQL_CALC_FOUND_ROWS in the SELECT statement. In this case, you can invoke the FOUND_ROW() function to retrieve the actual number of rows that are returned by the SELECT statement that does not have the LIMIT clause.

Example

```
mysql> SELECT SQL_CALC_FOUND_ROWS * FROM tbl_name
    -> WHERE id > 100 LIMIT 10;
mysql> SELECT FOUND_ROWS();
```

The second SELECT statement returns a number that indicates how many rows are returned by the first SELECT statement if the first statement does not have the LIMIT clause. If the preceding SELECT statements do not use the SQL_CALC_FOUND_ROWS option, the results of FOUND_ROWS() may differ based on whether the LIMIT clause is used.

For the SELECT SQL_CALC_FOUND_ROWS statement in the preceding example, the returned value of the FOUND_ROWS() function is valid only for a short period. The returned value becomes invalid when the statement that follows the SELECT SQL_CALC_FOUND_ROWS statement is executed. If you need to use the returned number of rows later, you must save the number.

Example

```
mysql> SELECT SQL_CALC_FOUND_ROWS * FROM ... ;
mysql> SET @rows = FOUND_ROWS();
```

If you use SQL_CALC_FOUND_ROWS in a query, the system calculates the number of rows in the full result set. This process requires less time than the process of running another query that does not use the LIMIT clause. This is because the result set in the former process does not need to be sent to the client.

Assume that you want to limit the number of rows that a query returns and do not want to run another query to retrieve the number of rows in the full result set. In this case, you can use SQL_CALC_FOUND_ROWS and FOUND_ROWS(). **For example**, you can use a web script for a paged display. The displayed information contains the links to pages for the other parts of the query results. You can use the FOUND_ROWS() function to determine the number of additional pages that are required to show the remaining results.

The implementation of SQL_CALC_FOUND_ROWS and FOUND_ROWS() in UNION queries is more complex than that in simple SELECT statements. This is because a UNION query may contain one or more LIMIT clauses. For example, you may use the LIMIT clauses in the SELECT statements of a UNION query, or use the clauses to limit the UNION results.

If SQL_CALC_FOUND_ROWS is used in a UNION query, the expected return result is the row count that is not limited by the global LIMIT clause. If you need to use SQL_CALC_FOUND_ROWS in a UNION query, ensure that the following requirements are met:

- The SQL_CALC_FOUND_ROWS keyword must appear in the first SELECT statement of the UNION query.
- The value of FOUND_ROWS() is exact only if UNION ALL is used. If UNION without ALL is used, duplicate removal occurs and the value of FOUND_ROWS() is only approximate.
- If the UNION query does not contain the LIMIT clauses, SQL_CALC_FOUND_ROWS is ignored. In this case, the query returns the number of rows in the temporary table that is created to process the UNION query.

LAST_INSERT_ID()

Declaration

```
last_insert_id()
```

Description

This function returns the auto-increment field value that is latest inserted in the current session. If you insert multiple rows into the table in the latest operation, the LAST_INSERT_ID() function returns the auto-increment field value of the first row.

Example

```
mysql>select LAST_INSERT_ID();
+------------------+
| LAST_INSERT_ID() |
+------------------+
|                5 |
+------------------+
1 row in set (0.00 sec)
```

17.1.4.3.5. Other functions

COALESCE

Declaration

```
COALESCE(expr, expr, expr,...)
```

Description

This function evaluates expression values in sequence until the function finds a non-NULL value. Then, the function returns the non-NULL value. If the values of all the expressions are NULL, the function returns a NULL value.

All expressions must be in the same data type. If the data types of expressions are different, the system converts the data types into the same type in an implicit way.

Example

```
Oceanbase>SELECT COALESCE(NULL,NULL,3,4,5), COALESCE(NULL,NULL,NULL);
+---------------------------+--------------------------+
| COALESCE(NULL,NULL,3,4,5) | COALESCE(NULL,NULL,NULL) |
+---------------------------+--------------------------+
|                         3 |                     NULL |
+---------------------------+--------------------------+
1 row in set (0.00 sec)
```

NVL

Declaration

```
NVL(str1,replace_with)
```

Description

This function returns the replace_with value if the str1 value is NULL.

If you set str1 to NULL, the function returns a value that you set for the replace_with parameter. This helps you retrieve complete outputs. In most cases, the str1 value is a column name. No limits are placed on the values of replace_with. For example, you can specify hard-coded values, references to other columns, or expressions for the replace_with parameter.

Example

```
Oceanbase>SELECT NVL(NULL, 0), NVL(NULL, 'a');
+--------------+----------------+
| NVL(NULL, 0) | NVL(NULL, 'a') |
+--------------+----------------+
|            0 | a              |
+--------------+----------------+
1 row in set (0.00 sec)
```

SLEEP

Declaration

```
SLEEP(duration)
```

Description

This function suspends an SQL query for a specified duration that is measured in seconds. The function returns 0 after the suspension ends.

If only the SLEEP function is run in a query that is not interrupted, the function returns 0.

If only the SLEEP function is run in a query that is interrupted, the function returns 1 and does not return an error code.

If the SLEEP function is part of a query and the query is interrupted during the suspension, the function returns error code 1317.

Example

```
mysql> SELECT SLEEP(1000);
+------------------+
| SLEEP(1000) |
+------------------+
|          0 |
+------------------+

mysql> SELECT SLEEP(1000);
+------------------+
| SLEEP(1000) |
+------------------+
|          1 |
+------------------+

mysql> SELECT 1 FROM t1 WHERE SLEEP(1000);
ERROR 1317 (70100): Query execution was interrupted
```

Full-text search functions

Declaration

```
MATCH (col1,col2,...) AGAINST (expr [search_modifier])
search_modifier:
{
    IN NATURAL LANGUAGE MODE
    | IN NATURAL LANGUAGE MODE WITH QUERY EXPANSION
    | IN BOOLEAN MODE
    | WITH QUERY EXPANSION
}
```

Description

ApsaraDB for OceanBase V1.0 supports full-text search functions. You can use such functions to query full-text indexes. Before you use full-text search functions, ensure the following requirements are met.

- Full-text indexes are created on the columns that are specified in the MATCH(col1,col2,...) function. ApsaraDB for OceanBase supports only FULLTEXT CTXCAT indexes.
- The MATCH(col1,col2,...) function must include the columns that are specified for the FULLTEXT CTXCAT indexes. For example, you can create the FULLTEXT INDEX(c1,c2,c3) CTXCAT(c2,c3) index. In this scenario, the retrieved data can be matched only if the function is MATCH(c2,c3).
- You can specify the search mode for the full-text search function by using the search_modifier parameter. ApsaraDB for OceanBase supports only two modes: NATURAL LANGUAGE MODE and BOOLEAN MODE. The default mode is NATURAL LANGUAGE MODE.

By default or when you use IN NATURAL LANGUAGE MODE, the MATCH...AGAINST statement uses NATURAL LANGUAGE MODE for full-text searches. In NATURAL LANGUAGE MODE, the AGAINST clause accepts a search string and searches in the index by comparing character sets. For each row in the table, the MATCH function returns the correlation between the string and the row data. The correlation represents the similarity between the string and the text in the data table. By default, string columns created in ApsaraDB for OceanBase are case-insensitive. Therefore, keywords for full-text searches are case-insensitive. If you need to run case-sensitive full-text searches, you can specify case-sensitive data types for the columns on which full-text indexes are created. For example, you can specify the data type of the columns as UTF8MB4_BIN. If the MATCH...AGAINST function is used in the WHERE clause, MATCH is used to filter data that is irrelevant to the keywords in the function. ApsaraDB for OceanBase supports only MATCH...AGAINST=0 and MATCH...AGAINST>0. MATCH...AGAINST=0 indicates that no data matches the keywords and MATCH...AGAINST>0 indicates that at least one keyword is matched. The AGAINST parameter can accept multiple keywords. The keywords are separated with spaces, which indicates the logical OR relationship. If one of the keywords matches the text, this is considered as a match.

ApsaraDB for OceanBase can run full-text searches in BOOLEAN mode by using the IN BOOLEAN MODE keyword. In this mode, some special operators in front of the keywords have special semantic meanings. Examples

```
SELECT * FROM t1 WHERE MATCH (a, b) AGAINST ('Chrysanthemum Jasmine' IN BOOLEAN MODE);
+----+------------+------------+
| id | a            | b            |
+----+------------+------------+
| 1 | Alipay        | Chrysanthemum tea        |
| 2 | Taobao          | Jasmine        |
+----+------------+------------+

SELECT * FROM t1 WHERE MATCH (a, b)
    AGAINST ('+Chrysanthemum -Jasmine' IN BOOLEAN MODE);
+----+------------+------------+
| id | a            | b            |
+----+------------+------------+
| 1 | Alipay        | Chrysanthemum tea        |
+----+------------+------------+
```

The BOOLEAN full-text search in ApsaraDB for OceanBase supports the following operators: Plus signs (+) represent logical AND relationships. The search result must also include the keywords that are preceded by a plus sign (+). Hyphens (-) represent logical NOT relationships. The search result cannot include the keywords that are preceded by a hyphen (-).

```
If no operator is specified, the logical OR relationship is applied to the specified keywords. This i
ndicates that the search result includes at least one of the specified keywords that are not preceded
by an operator.
```

In BOOLEAN mode, take note of this point: All operators must be placed in front of the keywords, and operators behind the keywords are ignored. For example, the plus sign (+) in "+Chrysanthemum" has a semantic meaning, but is ignored in "Chrysanthemum+". Operators and keywords must be connected in a close way and cannot be separated by other characters. Otherwise, the operators are ignored. For example, the plus sign (+) in "+ Chrysanthemum" is ignored.

17.1.4.4. Queries and subqueries

17.1.4.4.1. Overview

An SQL query refers to the method used to obtain data from a database. An SQL query can be used in conjunction with conditional clauses such as WHERE and ordering clauses such as ORDER BY to obtain query results. A subquery is a query that is nested within an external query. The external query is also called the parent query or the outer query. The result of a subquery is passed back as input to the parent query or outer query. The parent query uses this value in the computation to determine the final output. SQL allows multiple levels of nested queries. This means one subquery can also have other subqueries nested. At the same time, subqueries can appear in various clauses in SQL statements, such as SELECT, FROM, and WHERE statements.

Subqueries

In databases, subqueries can be divided into dependent subqueries and independent subqueries. A dependent subquery is a subquery whose execution depends on the variables of an external query. In most case, such subqueries are calculated multiple times. An independent subquery is a subquery whose execution does not depend on the variables of an external query. Such subqueries are calculated only once. The following figure shows an independent subquery and a dependent subquery.

```
OceanBase (root@test)> create table t1(a int primary key, b int, c int);
Query OK, 0 rows affected (0.70 sec)
OceanBase (root@test)> create table t2(a int primary key, b int, c int);
Query OK, 0 rows affected (0.92 sec)
-- An independent subquery
OceanBase (root@test)> select * from t1 where t1.a in (select t2.a from t2);
Empty set (0.22 sec)
-- A dependent subquery. The outer query variable t1.b is used in the subquery.
OceanBase (root@test)> select * from t1 where t1.a in (select t2.a from t2 where t2.b = t1.b);
Empty set (0.05 sec)
```

Subquery unnesting

Subquery unnesting is a strategy to optimize databases. This strategy merges the subquery into the body of the outer query by converting some subqueries into equivalent multi-table JOIN operations. One of the benefits of this strategy is that the optimizer can select appropriate access paths, join methods, and join orders so that the query hierarchy is reduced as much as possible. The following figure shows an example of subquery unnesting, where the subquery is rewritten into a JOIN statement.

```
OceanBase (root@test)> create table t1(a int primary key, b int, c int);
Query OK, 0 rows affected (0.70 sec)
OceanBase (root@test)> create table t2(a int primary key, b int, c int);
Query OK, 0 rows affected (0.92 sec)
--- Dependent subqueries are unnested and rewritten into JOIN statements.
OceanBase (root@test)> explain select * from t1 where t1.a in (select t2.b from t2 where t2.c = t1.c)
;
| ======================================
|ID|OPERATOR      |NAME|EST. ROWS|COST|
--------------------------------------
|0 |HASH SEMI JOIN|    |1        |2924|
|1 | TABLE SCAN   |t1  |1000     |455 |
|2 | TABLE SCAN   |t2  |1000     |455 |
======================================

Outputs & filters:
-------------------------------------
  0 - output([t1.a], [t1.b], [t1.c]), filter(nil),
      equal_conds([t1.a = t2.b], [t2.c = t1.c]), other_conds(nil)
  1 - output([t1.c], [t1.a], [t1.b]), filter(nil),
      access([t1.c], [t1.a], [t1.b]), partitions(p0)
  2 - output([t2.c], [t2.b]), filter(nil),
      access([t2.c], [t2.b]), partitions(p0)
```

17.1.4.4.2. JOIN operations

The JOIN statement is used to combine two or more tables based on the JOIN conditions. The sets that are created by executing JOIN statements can be saved or used as tables. The JOIN statement is used to combine the properties of two tables based on the values in the tables. The JOIN types in a database include inner join, outer join, semi join, and anti join. Semi join and anti join are obtained by rewriting subqueries. SQL itself does not support the anti join or semi join expression.

JOIN conditions

JOIN conditions can be divided into equi join and non-equi join. Examples of equi join and non-equi join: t1.a = t2.b and t1.a < t2.b. Compared with non-equi join, one of the benefits of equi join is that it allows you to use efficient JOIN algorithms such as hash join and merge-sort join.

Self-join

Self join indicates that a table is joined with itself. The following figure shows an example of a self join.

```
OceanBase (root@test)> create table t1(a int primary key, b int, c int);
Query OK, 0 rows affected (0.70 sec)
--- Example of a self join
OceanBase (root@test)> select * from t1 as ta, t1 as tb where ta.b = tb.b
```

Inner join

Inner join is one of the most basic JOIN operations in a database. Inner join creates a new table that combines the columns of two tables (such as Table A and Table B) based on the JOIN conditions. A query compares each row in Table A with each row in Table B and finds combinations that satisfy the JOIN conditions. If a row in Table A and a row in Table B satisfy the JOIN conditions, the two rows are merged into one row in the new table. The result set of the JOIN operation is the outcome of first obtaining the Cartesian product of two tables and then returning all rows that match the JOIN conditions. A Cartesian product is also known as a cross join, which returns a result table where each row from Table A is combined with each row from Table B.

Outer join

Outer join does not require each record of the two joined tables to find a matching record in the other table. If all the records in a table, including non-matching records, are retained in the joined tables, this table is called a preserved table. Outer join can be divided into left outer join, right outer join, and full outer join based on whether to retain rows from the left, right, or both tables. If a row in the left table is not found in the right table, left outer join automatically returns NULL in the right table. If a row in the right table is not found in the left table, right outer join automatically returns NULL in the left table. If rows in the left and right tables do not match, full outer join automatically returns NULL in the tables.

Semi join

When Table A and Table B are left semi-joined, the operation only returns all rows in Table A that can be found in Table B. Right semi join only returns all rows in Table B that can be found in Table A. Semi join can only be obtained by subquery unnesting, as shown in the following figure.

```
OceanBase (root@test)> create table t1(a int primary key, b int, c int);
Query OK, 0 rows affected (0.70 sec)
OceanBase (root@test)> create table t2(a int primary key, b int, c int);
Query OK, 0 rows affected (0.92 sec)
--- A dependent subquery is unnested and rewritten as a semi join.
OceanBase (root@test)> explain select * from t1 where t1.a in (select t2.b from t2 where t2.c = t1.c)
;
| ======================================
|ID|OPERATOR      |NAME|EST. ROWS|COST|
--------------------------------------
|0 |HASH SEMI JOIN|    |1        |2924|
|1 | TABLE SCAN   |t1  |1000     |455 |
|2 | TABLE SCAN   |t2  |1000     |455 |
======================================

Outputs & filters:
-------------------------------------
  0 - output([t1.a], [t1.b], [t1.c]), filter(nil),
      equal_conds([t1.a = t2.b], [t2.c = t1.c]), other_conds(nil)
  1 - output([t1.c], [t1.a], [t1.b]), filter(nil),
      access([t1.c], [t1.a], [t1.b]), partitions(p0)
  2 - output([t2.c], [t2.b]), filter(nil),
      access([t2.c], [t2.b]), partitions(p0)
```

Anti join

When Table A and Table B are left anti-joined, the operation only returns all rows in Table A that cannot be found in Table B. Right anti join only returns all rows in Table B that cannot be found in Table A. Similar to semi join, anti join can be obtained only by subquery unnesting, as shown in the following figure.

```
OceanBase (root@test)> create table t1(a int primary key, b int, c int);
Query OK, 0 rows affected (0.70 sec)
OceanBase (root@test)> create table t2(a int primary key, b int, c int);
Query OK, 0 rows affected (0.92 sec)
--- A dependent subquery is rewritten as an anti join.
OceanBase (root@test)> explain select * from t1 where t1.a not in (select t2.b from t2 where t2.c = t
1.c);
| ======================================
|ID|OPERATOR      |NAME|EST. ROWS|COST|
--------------------------------------
|0 |HASH ANTI JOIN|    |995      |3262|
|1 | TABLE SCAN   |t1  |1000     |455 |
|2 | TABLE SCAN   |t2  |1000     |455 |
======================================

Outputs & filters:
-------------------------------------
  0 - output([t1.a], [t1.b], [t1.c]), filter(nil),
      equal_conds([t2.c = t1.c]), other_conds([t1.a = t2.b OR (T_OP_IS, t2.b, NULL, 0)])
  1 - output([t1.c], [t1.a], [t1.b]), filter(nil),
      access([t1.c], [t1.a], [t1.b]), partitions(p0)
  2 - output([t2.c], [t2.b]), filter(nil),
      access([t2.c], [t2.b]), partitions(p0)
```

17.1.4.4.3. Sets

The set operations on a database can combine the results of multiple queries into a result set. The following list describes the set operations:

- UNION
- INTERSECT
- In ApsaraDB for OceanBase, EXCEPT and MINUS are supported and have the same semantics. The queries of a set operation must result in the same number of columns and compatible data types. For UNION, the available UNION attributes are ALL and DISTINCT or UNIQUE. ALL indicates that duplicated rows are allowed in the set. DISTINCT or UNIQUE indicates that duplicated rows are not allowed in the set. For other set operations, you can use the DISTINCT keyword only, and cannot use the ALL keyword. By default, DISTINCT is used for all set operations. ApsaraDB for OceanBase supports the ORDER BY and LIMIT clauses for set operations. Other clauses are not allowed, as shown in the following figure.

```
OceanBase (root@test)> create table t1(a int primary key, b int, c int);
Query OK, 0 rows affected (0.16 sec)
OceanBase (root@test)> create table t2(a int primary key, b int, c int);
Query OK, 0 rows affected (0.10 sec)
--The ORDER BY and LIMIT clauses are supported in the UNION statement.
OceanBase (root@test)> (select * from t1 union all select * from t2) order by a limit 10;
Empty set (0.02 sec)
--Other clauses such as GROUP BY are not supported in the UNION statement, except the ORDER BY and LI
MIT clauses.
OceanBase (root@test)> OceanBase (root@test)> (select * from t1 union all select * from t2) group by
a limit 10;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your O
ceanBase version for the right syntax to use near 'OceanBase (root@test)> (select * from t1 union all
select * from t2) group by a ' at line 1
```

Example of the UNION statement

This example returns all the unique rows in t1 and t2.

```
OceanBase (root@test)> create table t1(a int, b int, c int);
Query OK, 0 rows affected (0.12 sec)
OceanBase (root@test)> create table t2(a int, b int, c int);
Query OK, 0 rows affected (0.11 sec)
OceanBase (root@test)> insert into t1 values (1,1,1),(2,2,2),(3,3,3);
Query OK, 3 rows affected (0.07 sec)
Records: 3  Duplicates: 0  Warnings: 0
OceanBase (root@test)> insert into t2 values (2,2,2),(3,3,3),(4,4,4);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
OceanBase (root@test)> select * from t1 union select * from t2;
+------+------+------+
| a    | b    | c    |
+------+------+------+
|    1 |    1 |    1 |
|    2 |    2 |    2 |
|    3 |    3 |    3 |
|    4 |    4 |    4 |
+------+------+------+
4 rows in set (0.01 sec)
```

Example of the UNION ALL statement

This example returns all rows in t1 and t2 including duplicate rows.

```
OceanBase (root@test)> create table t1(a int, b int, c int);
Query OK, 0 rows affected (0.12 sec)
OceanBase (root@test)> create table t2(a int, b int, c int);
Query OK, 0 rows affected (0.11 sec)
OceanBase (root@test)> insert into t1 values (1,1,1),(2,2,2),(3,3,3);
Query OK, 3 rows affected (0.07 sec)
Records: 3  Duplicates: 0  Warnings: 0
OceanBase (root@test)> insert into t1 values (2,2,2),(3,3,3),(4,4,4);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
OceanBase (root@test)> select * from t1 union all select * from t2;
+------+------+------+
| a    | b    | c    |
+------+------+------+
|    1 |    1 |    1 |
|    2 |    2 |    2 |
|    3 |    3 |    3 |
|    2 |    2 |    2 |
|    3 |    3 |    3 |
|    4 |    4 |    4 |
+------+------+------+
6 rows in set (0.02 sec)
```

Example of the INTERSECT statement

This example returns rows that appear in both t1 and t2 without duplicate rows.

```
OceanBase (root@test)> create table t1(a int, b int, c int);
Query OK, 0 rows affected (0.12 sec)
OceanBase (root@test)> create table t2(a int, b int, c int);
Query OK, 0 rows affected (0.12 sec)
OceanBase (root@test)> insert into t1 values (1,1,1),(2,2,2),(3,3,3);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
OceanBase (root@test)> insert into t2 values (2,2,2),(3,3,3),(3,3,3),(4,4,4);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
OceanBase (root@test)> select * from t1 intersect select * from t2;
+------+------+------+
| a    | b    | c    |
+------+------+------+
|    2 |    2 |    2 |
|    3 |    3 |    3 |
+------+------+------+
2 rows in set (0.01 sec)
```

Example of the EXCEPT or MINUS statement

This example returns rows that appear in t1 but do not appear in t2 without duplicate rows.

```
OceanBase (root@test)> create table t1(a int, b int, c int);
Query OK, 0 rows affected (0.12 sec)
OceanBase (root@test)> create table t2(a int, b int, c int);
Query OK, 0 rows affected (0.12 sec)
OceanBase (root@test)> insert into t1 values (1,1,1),(2,2,2),(3,3,3);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
OceanBase (root@test)> insert into t2 values (2,2,2),(3,3,3),(3,3,3),(4,4,4);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
OceanBase (root@test)> select * from t1 except select * from t2;
+------+------+------+
| a    | b    | c    |
+------+------+------+
|    1 |    1 |    1 |
+------+------+------+
1 row in set (0.02 sec)
```

17.1.4.5. SQL statements

17.1.4.5.1. General syntax

Constants

- `INT_VALUE` : integer constants. For example, `123` .
- `DECIMAL_VALUE` : fixed-point constants. For example, `123.456` .
- `STR_VALUE` : string constants. For example, `abc` .
- `NULL` : a NULL constant.
- `STORAGE_SIZE` : storage size constants. The default unit is bytes for integer constants. M and G are units for string constants. For example, `1024` or `500M` .

```
const_value:
      INT_VALUE
    | DECIMAL_VALUE
    | STR_VALUE
    | NULL
    | STORAGE_SIZE

STORAGE_SIZE:
      INT_VALUE
    | 'INT_VALUE[M|G]'
```

Character sets

```
charset:
      default_charset
    | column_charset

default_charset:
    [DEFAULT] {CHARSET | CHARACTER SET} [=] {UTF8 | UTF8MB4 | BINARY}

column_charset:
    {CHARSET | CHARACTER SET} {UTF8 | UTF8MB4 | BINARY}
```

Collations

```
collate:
      default_collate
    | column_collate

default_collate:
    [DEFAULT] COLLATE [=] {UTF8MB4_GENERAL_CI | UTF8MB4_BIN | BINARY}

column_collate:
    COLLATE {UTF8MB4_GENERAL_CI | UTF8MB4_BIN | BINARY}
```

Data types

```
data_type:
      TINYINT[(precision)] [UNSIGNED] [ZEROFILL]
    | SMALLINT[(precision)] [UNSIGNED] [ZEROFILL]
    | MEDIUMINT[(precision)] [UNSIGNED] [ZEROFILL]
    | INT[(precision)] [UNSIGNED] [ZEROFILL]
    | INTEGER[(precision)] [UNSIGNED] [ZEROFILL]
    | BIGINT[(precision)] [UNSIGNED] [ZEROFILL]
    | FLOAT[(precision, scale)] [UNSIGNED] [ZEROFILL]
    | DOUBLE[(precision, scale)] [UNSIGNED] [ZEROFILL]
    | DECIMAL[(precision [, scale])] [UNSIGNED] [ZEROFILL]
    | NUMERIC[(precision [, scale])] [UNSIGNED] [ZEROFILL]
    | DATETIME[(scale)]
    | TIMESTAMP[(scale)]
    | DATE
    | TIME[(scale)]
    | YEAR
    | VARCHAR(length) column_charset column_collate
    | VARBINARY(length)
    | CHAR[(length)] column_charset column_collate
    | BINARY[(length)]
    | TINYTEXT column_charset column_collate
    | TINYLOB
    | TEXT[(length)] column_charset column_collate
    | BLOB[(length)]
    | MEDIUMTEXT column_charset column_collate
    | MEDIUMBLOB
    | LONGTEXT column_charset column_collate
    | LONGBLOB

precision | scale | length:
    INT_VALUE
```

SQL statement attributes

- Object names

```
tenant_name | pool_name | unit_name | zone_name | region_name:
    STR_VALUE

database_name | table_name | table_alias_name | column_name| column_alias_name  | partition_name | su
bpartition_name:
    STR_VALUE

index_name | view_name | object_name | constraint_name | tablegroup_name:
    STR_VALUE

outline_name | user_name:
    STR_VALUE

table_factor:
    [[database_name].] table_name

column_factor:
    [table_factor.] column_name
```

- Expressions

```
expression:
      const_value
    | column_factor
    | operator_expression
    | function_expression
```

- Comments

```
comment:
    COMMENT 'comment_text'

comment_text:
    STR_VALUE
```

Distributed architecture attributes

- `PRIMARY_ZONE` : the distribution strategy of the replica leader.

```
primary_zone:
    PRIMARY_ZONE [=] zone_name
```

- `ZONE_LIST` : the resource distribution strategy of the tenants.

```
zone_list:
    ZONE_LIST [=] (zone_name [, zone_name ...])
```

- `REPLICA_NUM` : the number of data replicas.

```
replica_num:
    REPLICA_NUM [=] INT_VALUE
```

- `TABLEGROUP` : the distribution strategy of the replica leader for multiple data sets.

```
tablegroup:
      default_tablegroup
    | table_tablegroup

default_tablegroup:
    DEFAULT TABLEGROUP [=] {tablegroup_name | NULL}

table_tablegroup:
    TABLEGROUP [=] {tablegroup_name | NULL}
```

Data storage attributes

- `BLOCK_SIZE` : the micro-block size for object storage.

```
block_size:
    BLOCK_SIZE [=] INT_VALUE
```

- `COMPRESSION` : the compression algorithm for object storage.

```
compression:
    COMPRESSION [=] {NONE | LZ4_1.0 | LZ0_1.0 | SNAPPY_1.0 | ZLIB_1.0}
```

- `PCTFREE` : the ratio of idle space reserved in a macro-block for object storage.

```
pctfree:
    PCTFREE [=] INT_VALUE
```

- `TABLET_SIZE` : the minimum size of each data shard for parallel compaction in a single task.

```
tablet_size:
    TABLET_SIZE [=] INT_VALUE
```

17.1.4.5.2. ALTER DATABASE

Description

You can execute the ALTER DATABASE statement to modify the attributes of a database.

Syntax

```
alter_database_stmt:
        ALTER DATABASE [database_name] [SET] alter_specification_list;

alter_specification_list:
    alter_specification [alter_specification ...]

alter_specification:
    [DEFAULT] {CHARACTER SET | CHARSET} [=] charset_name
  | [DEFAULT] COLLATE [=] collation_name
  | REPLICA_NUM [=] int_num
  | PRIMARY_ZONE [=] zone_name
  | {READ ONLY | READ WRITE}
    | DEFAULT TABLEGROUP [=] {NULL | table_group_name}
```

Parameters

Parameter	Description
database_name	The name of the database for which you want to modify attributes. The current default database is modified if you leave this parameter empty.
CHARSET charset_name	The character set that you want to modify.
COLLATE collation_name	The collation.
REPLICA_NUM int_num	The number of replicas.
PRIMARY_ZONE zone_name	The primary zone.
READ ONLY | READ WRITE	The attribute of the database. Valid values: READ ONLY and READ WRITE.
DEFAULT TABLEGROUP table_group_name	The default table group of the database. A NULL value indicates that no default table group is specified for the database.

Examples

- Execute the following statements to modify database test2. Set the character set to UTF8MB4 and the collation to UTF8MB4_BIN. Set the database attribute to READ WRITE.

```
OceanBase(admin@test)>alter database test2 DEFAULT CHARACTER SET UTF8MB4;
Query OK, 0 rows affected (0.03 sec)

OceanBase(admin@test)>alter database test2 DEFAULT COLLATE UTF8MB4_BIN;
Query OK, 0 rows affected (0.03 sec)

OceanBase(admin@test)>alter database test2 READ WRITE;
Query OK, 0 rows affected (0.02 sec)
```

17.1.4.5.3. ALTER OUTLINE

Description

You can execute the ALTER OUTLINE statement to modify outlines and configure throttling rules for SQL queries. This statement can be used only for outlines that are created based on SQL texts.

Syntax

```
ALTER OUTLINE outline_name ADD stmt [ TO target_stmt ]
```

Parameters

Parameter	Description
outline_name	The name of the outline that you want to modify.
stmt	The original data manipulation language (DML) statement that contains a hint and parameters.
TO target_stmt	You can leave TO target_stmt empty if the following requirements are met: The parameterized SQL statement is equivalent to the stmt value without the hint. In this case, the system generates an execution plan for the parameterized SQL statement that has the hint in stmt. To generate a fixed execution plan for a statement that contains a hint, specify the original SQL statement as TO target_stmt. **Notice** Make sure TO target_stmt you specify is the same as the stmt value without the hint.

Examples

- Execute the ALTER OUTLINE statement to add throttling rules.

```
OceanBase (root@oceanbase)> alter outline ol_1 add select /*+max_concurrent(1)*/ * from t1 where c1 =
1 and c2 = ? ;
OceanBase (root@oceanbase)> alter outline ol_1 add select /*+max_concurrent(1)*/ * from t1 where c1 =
? and c2 = 1;
```

- Execute the ALTER OUTLINE statement to add an execution plan.

```
OceanBase (root@oceanbase)> create outline ol_2 on select /*+max_concurrent(1)*/ * from t1,t2 where t
1.c1 = 1;
OceanBase (root@oceanbase)> alter outline ol_2 add select /*+use_nl(t2)*/ * from t1,t2 where t1.c1 =
1;
```

Notes

- One outline_name argument is associated with only one execution plan. If an execution plan is added by executing CREATE OUTLINE, you cannot execute ALTER OUTLINE to add an execution plan again.
- When you execute the ALTER OUTLINE statement, you can specify only a throttling rule or an execution plan for each execution. The similar rule applies to the CREATE OUTLINE statement.

17.1.4.5.4. ALTER RESOURCE POOL

Description

You can execute the ALTER RESOURCE POOL statement to modify a resource pool.

Syntax

```
ALTER RESOURCE POOL poolname
  UNIT [=] unitname,
  UNIT_NUM [=] unitnum,
  ZONE [=] ('zone' [, 'zone' …]) ;
```

Parameters

Parameter	Description
UNIT	The name of the resource unit that you want to use.
UNIT_NUM	The number of resource units that you want to create in a zone. The value must be smaller than the number of OBServers in the zone.
ZONE_LIST	The zones in which you want to create resource units.

Examples

- You cannot modify multiple attributes for a resource pool at a time, such as UNIT, UNIT_NUM, and ZONE_LIST. Otherwise, the system returns an error.

```
OceanBase(admin@test)> ALTER RESOURCE POOL pool1 unit='unit2', unit_num=1, zone_list=('zone1');
ERROR 1235 (0A000): alter unit_num, resource_unit, zone_list in one cmd not supported
```

- Modify the UNIT attribute for a resource pool.

```
OceanBase(admin@test)> ALTER RESOURCE POOL pool1 unit='unit2';
Query OK, 0 rows affected (0.00 sec)
```

17.1.4.5.5. ALTER RESOURCE UNIT

Description

You can execute the ALTER RESOURCE UNIT statement to modify a resource unit.

Syntax

```
ALTER RESOURCE UNIT unitname
  MAX_CPU [=] cpunum,
  MAX_MEMORY [=] memsize,
  MAX_IOPS [=] iopsnum,
  MAX_DISK_SIZE [=] disksize,
  MAX_SESSION_NUM [=] sessionnum,
  [MIN_CPU [=] cpunum,]
  [MIN_MEMORY [=] memsize,]
  [MIN_IOPS [=] iopsnum] ;
```

Parameters

Parameter	Description
MAX_CPU	The maximum number of CPUs.
MAX_MEMORY	The maximum memory size. Valid values: [1073741824,+∞). The unit is bytes. The minimum value 1073741824 bytes is equivalent to 1 GB.
MAX_IOPS	The maximum IOPS. Valid values: [128,+∞).
MAX_DISK_SIZE	The maximum disk capacity. Valid values: [536870912,+∞). The unit is bytes. The minimum value 536870912 bytes is equivalent to 512 MB.
MAX_SESSION_NUM	The maximum number of sessions. Valid values: [64,+∞).
MIN_CPU	The minimum number of CPUs.
MIN_MEMORY	The minimum memory size.
MIN_IOPS	The minimum IOPS.

Examples

- Modify resource unit unit1 by changing the maximum number of CPUs to two and memory size to 2 GB.

```
OceanBase(admin@test)> ALTER RESOURCE UNIT unit1 max_cpu 2, max_memory '2G';
Query OK, 0 rows affected (0.02 sec)
```

17.1.4.5.6. ALTER SYSTEM

You can execute the ALTER SYSTEM statement to send commands to the ApsaraDB for OceanBase system to perform specified operations.

BOOTSTRAP

Description

You can use this statement to bootstrap an ApsaraDB for OceanBase cluster.

Syntax

```
alter_system_bootstrap_stmt:
    ALTER SYSTEM BOOTSTRAP opt_cluster_type region_zone_server_list;

opt_cluster_type:
    [CLUSTER cluster_role]

cluster_role:
    PRIMARY | STANDBY

region_zone_server_list:
    region_zone_server [, region_zone_server ...]

region_zone_server:
    [region] zone server

region:
    REGION [=] region_name

zone:
    ZONE [=] zone_name

server:
    SERVER [=] ip_port

ip_port:
    'STR_VALUE:INT_VALUE'
```

Parameters

To bootstrap the system, specify the RootService nodes. Use commas (,) to separate multiple RootService nodes.

Parameter	Description
region_name	The region where the RootService node is deployed. Specify this parameter if the cluster is deployed across zones in multiple regions.
zone_name	The zone to which the RootService node belongs.
ip_port	The IP address and port number of the RootService node.

Parameter	Description
PRIMARY | STANDBY	Specifies whether to run the primary or secondary database. Use the PRIMARY or STANDBY keyword to specify the database that is used when the system is started. By default, if you do not use PRIMARY or STANDBY, the system runs the primary database.

Examples

- Specify a RootService node.

```
ALTER SYSTEM BOOTSTRAP ZONE 'zone1' SERVER '10.218.248.178:55410';
```

- Separate multiple RootService nodes with commas (,).

```
ALTER SYSTEM BOOTSTRAP ZONE 'zone1' SERVER '172.24.65.24:55410', ZONE 'zone2'
SERVER '172.24.65.114:55410';
```

- Run a secondary database.

```
ALTER SYSTEM BOOTSTRAP CLUSTER STANDBY ZONE 'zone1' SERVER '10.218.248.178:55410';
```

JOB

Description

You can use this statement to trigger a background job. You can configure parameters to specify a job.

Syntax

```
alter_system_job_stmt:
    ALTER SYSTEM RUN JOB job_name
    [zone | server];
```

Parameters

Parameter	Description

Parameter	Description
JOB job_name	The name of the job. If special characters are included, the job name must be enclosed in single quotation marks ('). If special characters are not included, you can determine whether to use single quotation marks (') as needed. ApsaraDB for OceanBase supports the following jobs: • check_partition_table: Check and delete partitioned tables on an OBServer. • root_inspection: Trigger a self-check job on a RootService node.
zone | server	Specifies whether to run the job in a specified zone or on a specified server.

Examples

- Trigger a self-check job on a RootService node.

```
ALTER SYSTEM RUN JOB "root_inspection";
```

MERGE

Description

You can use this statement to trigger major or minor freeze operations.

Syntax

```
alter_system_merge_stmt:
    ALTER SYSTEM merge_action;

merge_action:
      MAJOR FREEZE
    | MINOR FREEZE
      [tenant_list | replica] [server_list] [zone]
    | START MERGE
      zone
    | {SUSPEND | RESUME} MERGE
      [zone]
    | CLEAN MERGE ERROR

tenant_list:
    TENANT [=] (tenant_name_list)

tenant_name_list:
    tenant_name [, tenant_name ...]

replica:
    PARTITION_ID [=] 'partition_id%partition_count@table_id'

server_list:
    SERVER [=] ip_port_list
```

Parameters

Parameter	Description
MAJOR FREEZE	Trigger a daily major freeze operation.
MINOR FREEZE	Trigger a minor freeze operation.
START MERGE	Start a daily major freeze operation.
{SUSPEND | RESUME} MERGE	Suspend or resume a daily major freeze operation.
CLEAN MERGE ERROR	Clear errors that occur during major freeze operations.
tenant_name	The tenant on which minor freeze operations are performed.
PARTITION_ID	The partition in which minor freeze operations are performed.
SERVER	The OBServer on which minor freeze operations are performed.

Parameter	Description
zone	The zone in which major freeze operations are performed.

Examples

- Trigger a daily major freeze operation.

```
OceanBase(root@oceanbase)>alter system major freeze;
Query OK, 0 rows affected (0.06 sec)
```

PARAMETER

Description

You can use this statement to modify configuration items.

Syntax

```
alter_system_parameter_stmt:
    ALTER SYSTEM [SET]
    parameter_name = expression [SCOPE = {MEMORY | SPFILE | BOTH}] [COMMENT [=} 'text']
                                                                  {SERVER [=] 'ip:port' | ZONE [=] 'zone'};
```

Parameters

Parameter	Description
parameter_name	The name of the configuration item that you want to modify.
expression	The new value of the configuration item.
COMMENT 'text'	Add a comment about the modification. This parameter is optional. We recommend that you do not omit it.

Parameter	Description
SCOPE	The effective range of the modification. Valid values: • MEMORY: Only the configuration item in the memory is modified. The modification takes effect immediately but becomes invalid after the server is restarted. No configuration items support this option. • SPFILE: Only the configuration item in the configuration table is modified. The modification takes effect after the server is restarted. • BOTH: Both the configuration item in the configuration table and that in the memory are modified. The modification takes effect immediately and remains valid after the server is restarted. • The default value is BOTH. If you specify BOTH or MEMORY for a configuration item on which the modification cannot take effect immediately, the system returns an error.
SERVER	The configuration item of the specified server. This indicates that only the configuration item of the specified server is modified.
ZONE	The name of the zone. This indicates that the configuration item modification applies to the specified type of servers in the specified cluster. If you do not specify the zone name, the configuration item modification applies to the specified type of the servers in all clusters.

Note

To modify multiple system configuration items at a time, separate them with commas (,).

Execute the following statement to query system configuration items:

```
SHOW PARAMETERS [LIKE 'pattern' | WHERE expr];
```

Examples

- Modify the configuration item enable_sql_audit.

```
OceanBase(root@oceanbase)>show parameters like 'enable_sql_audit';
+------+----------+---------------+----------+-----------------+-----------+-------+----------------
----------------------------------------------------------------------------------------------------+-----
-----+---------+---------+-------------------+
| zone | svr_type | svr_ip        | svr_port | name            | data_type | value | info
| section  | scope   | source  | edit_level        |
+------+----------+---------------+----------+-----------------+-----------+-------+----------------
----------------------------------------------------------------------------------------------------+-----
-----+---------+---------+-------------------+
| z1   | observer | 11.11.111.111 |    19510 | enable_sql_audit | NULL      | True  | specifies wheth
er SQL audit is turned on. The default value is TRUE. Value: TRUE: turned on FALSE: turned off | OBSE
RVER | CLUSTER | DEFAULT | DYNAMIC_EFFECTIVE |
+------+----------+---------------+----------+-----------------+-----------+-------+----------------
----------------------------------------------------------------------------------------------------+-----
-----+---------+---------+-------------------+
1 rows in set (0.02 sec)

OceanBase(root@oceanbase)>alter system set enable_sql_audit = false scope = BOTH;
Query OK, 0 rows affected (0.05 sec)

OceanBase(root@oceanbase)>show parameters like 'enable_sql_audit';
+------+----------+---------------+----------+-----------------+-----------+-------+----------------
----------------------------------------------------------------------------------------------------+-----
-----+---------+---------+-------------------+
| zone | svr_type | svr_ip        | svr_port | name            | data_type | value | info
| section  | scope   | source  | edit_level        |
+------+----------+---------------+----------+-----------------+-----------+-------+----------------
----------------------------------------------------------------------------------------------------+-----
-----+---------+---------+-------------------+
| z1   | observer | 11.11.111.111 |    19510 | enable_sql_audit | NULL      | False | specifies wheth
er SQL audit is turned on. The default value is TRUE. Value: TRUE: turned on FALSE: turned off | OBSE
RVER | CLUSTER | DEFAULT | DYNAMIC_EFFECTIVE |
+------+----------+---------------+----------+-----------------+-----------+-------+----------------
----------------------------------------------------------------------------------------------------+-----
-----+---------+---------+-------------------+
1 rows in set (0.02 sec)
```

REFRESH

Description

You can use this statement to refresh database information, such as the schema, time zone, and caches.

Syntax

```
alter_system_refresh_stmt:
    ALTER SYSTEM refresh_action;

refresh_action:
      REFRESH SCHEMA
      [zone | server]
    | REFRESH TIME_ZONE_INFO
    | FLUSH cache_type CACHE
      [tenant_list] [GLOBAL]
    | FLUSH KVCACHE
      [tenant] [CACHE [=] cache_name]

cache_type:
      ALL
    | LOCATION
    | CLOG
    | ILOG
    | COLUMN_STAT
    | BLOCK_INDEX
    | BLOCK
    | ROW
    | BLOOM_FILTER
    | SCHEMA
    | PLAN

tenant:
    TENANT [=] tenant_name

cache_name:
```

Parameters

Parameter	Description
REFRESH SCHEMA	Refresh the schema. In normal cases, when the system executes a data definition language (DDL) statement, the RootService node notifies all OBServers to refresh the schemas. Assume that some OBServers are disconnected from the RootService node due to exceptions. In this case, you must manually refresh the schemas. You can refresh the schema for a single OBServer or refresh all the schemas in a cluster.
REFRESH TIME_ZONE_INFO	Refresh the local time zone of all servers in the cluster.
FLUSH cache_type CACHE	Clear the caches of the specified type.

Parameter	Description
FLUSH KVCACHE	Clear the KV caches. • If tenant and cache_name are specified, the system clears the specified KV cache under the specified tenant. • If only tenant is specified, the system clears all the KV cache under the specified tenant. • If you leave tenant and cache_name empty, the system clears all the KV cache under all tenants.

Examples

- Refresh the schema for a single OBServer:

```
ALTER SYSTEM REFRESH SCHEMA SERVER='172.24.65.24:55410';
```

- Refresh all schemas in a zone:

```
ALTER SYSTEM REFRESH SCHEMA ZONE='zone1';
```

REPLICA

Description

You can use this statement to migrate, replicate, or delete replicas. You can also change the replica type or switch replica roles.

Syntax

```
alter_system_replica_stmt:
    ALTER SYSTEM replica_action;

replica_action:
      SWITCH REPLICA
      {LEADER | FOLLOWER}
      {replica server | server [tenant_name] | zone [tenant_name]}
    | DROP REPLICA partition_id_desc
      replica server [create_timestamp] [zone] [FORCE]
    | {MOVE | COPY} REPLICA
      replica source destination
    | REPORT REPLICA partition_id_desc
         {zone | server}
      | RECYCLE REPLICA partition_id_desc
         {zone | server}
      | {ALTER | CHANGE | MODIFY} REPLICA
      replica server [set] REPLICA_TYPE = replica_type

source:
    SOURCE [=] 'ip:port'

destination:
    DESTINATION [=] 'ip:port'

partition_id_desc
    PARTITION_ID partition_id%partition_count@table_id

partition_idx | partition_count | table_id | task_id:
    INT_VALUE

create_timestamp:
    CREATE_TIMESTAMP [=] INT_VALUE

tenant_name_list:
    tenant_name [, tenant_name ...]

replica_type:
      {FULL | F}
    | {READONLY | R}
    | {LOGONLY | L}
```

Parameters

Parameter	Description
SWITCH REPLICA	Select a new replica leader.
DROP REPLICA	Delete a replica. To delete a replica on the specified OBServer, specify the following parameters: PARTITION_ID, SERVER, and CREATE_TIMESTAMP.

Parameter	Description
{MOVE | COPY} REPLICA	Migrate or replicate a replica. You must specify a source OBServer, a destination OBServer, and a partition ID.
REPORT REPLICA	Report replicas. This clause requires a single OBServer or all the OBServers in a zone to report replicas.
RECYCLE REPLICA	Recycle replicas that you no longer need.
{ALTER | CHANGE | MODIFY} REPLICA	Modify the replica attributes. You can modify the type of a specified replica. The following replica types are supported: FULL, READONLY, and LOGONLY. You can set REPLICA_TYPE to the full name or abbreviation of a valid replica type, such as F, R, or L. The value is case-insensitive.

Examples

- Migrate a replica.

```
ALTER SYSTEM MOVE REPLICA PARTITION_ID '0%4@1100611139403777'
SOURCE '172.24.65.24:55410'
DESTINATION '172.24.65.26:55410';
```

- Delete a replica.

```
ALTER SYSTEM DROP REPLICA PARTITION_ID '0%4@1100611139403777'
SERVER '172.24.65.26:55410';
```

- Modify the type of a replica.

```
ALTER SYSTEM CHANGE REPLICA PARTITION_ID '0%4@1100611139403777'
SERVER '172.24.65.26:55410';
CHANGE REPLICA_TYPE = 'L';
```

- Select a new replica leader.

```
ALTER SYSTEM SWITCH REPLICA LEADER PARTITION_ID '0%4@1100611139403777'
SERVER '172.24.65.26:55410';
```

ROOTSERVICE

Description

You can use this statement to change the role of a RootService node.

Syntax

```
alter_system_rootservice_stmt:
    ALTER SYSTEM SWITCH ROOTSERVICE {LEADER | FOLLOWER} {zone | server};
```

Parameters

Parameter	Description
LEADER | FOLLOWER	Set the role of a RootService node to LEADER or FOLLOWER.
zone | server	Change the role of a specified RootService node or a RootService node in a specified zone.

Examples

- Change the role of a RootService node in zone z1 to LEADER.

```
ALTER SYSTEM SWITCH ROOTSERVICE LEADER ZONE 'z1';
```

SERVER

Description

You can use this statement to maintain OBServers. For example, you can add, delete, start, or stop an OBServer.

Syntax

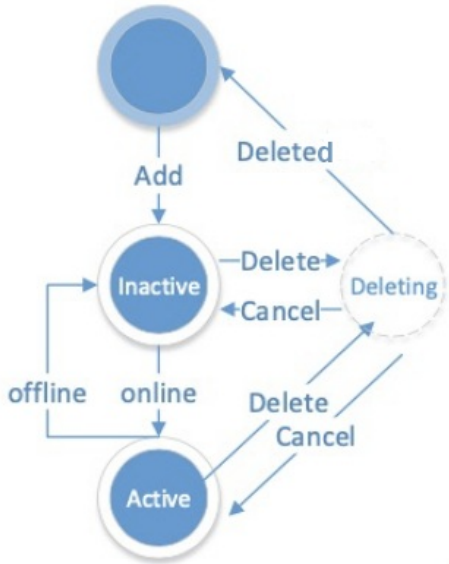
```
alter_system_server_stmt:
    ALTER SYSTEM server_action SERVER ip_port_list [zone];

server_action:
      ADD
    | DELETE
    | CANCEL DELETE
    | START
    | STOP
    | FORCE STOP

ip_port_list:
    ip_port [, ip_port ...]
```

Parameters

The following figure displays the OBServer status.



Parameter	Description
ip_port	The IP address and port number of the OBServer.
zone	If a zone is specified, the system checks whether the OBServer that you want to maintain is deployed in the specified zone.
ADD	Add an OBServer.
DELETE	Delete an OBServer.
CANCEL DELETE	Abort a DELETE operation on an OBServer.
START	Start an OBServer.
STOP	Stop an OBServer.
FORCE STOP	Forcibly stop an OBServer.

Examples

- Add an OBServer.

```
ALTER SYSTEM ADD SERVER '172.24.65.113:55410' ZONE 'zone1';
```

> **Notice**
>
> - You can execute the ALTER SYSTEM ADD SERVER or ALTER SYSTEM DELETE SERVER statement to add a server to or delete a server from the server list. Only servers on the server list can provide services.
> - When the ALTER SYSTEM DELETE SERVER statement is executed, the system selects a new replica leader and replicates replicas.
> - The ALTER SYSTEM DELETE SERVER statement is time-consuming. You can execute the ALTER SYSTEM CANCEL DELETE SERVER statement to abort this operation.

THROTTLE

Description

You can use this statement to configure throttling rules.

Syntax

```
alter_system_throttle_stmt:
    ALTER SYSTEM throttle_action;

throttle_action:
      ENABLE SQL THROTTLE [priority_option] [using_metric_option_list]
    | DISABLE SQL

priority_option:
    FOR PRIORITY <= INT_VALUE

using_metric_option_list:
    USING metric_option_list

metric_option_list:
    metric_option [metric_option ...]

metric_option:
      RT = {INT_VALUE | DECIMAL_VALUE}
    | CPU = {INT_VALUE | DECIMAL_VALUE}
    | IO = INT_VALUE
    | NETWORK = {INT_VALUE | DECIMAL_VALUE}
    | QUEUE_TIME = {INT_VALUE | DECIMAL_VALUE}
    | LOGICAL_READS = {INT_VALUE | DECIMAL_VALUE}
```

Parameters

Parameter	Description
ENABLE SQL	Enable throttling. You can configure the parameters to enable throttling based on the specified rules and metrics.
FOR PRIORITY	Configure the priority to throttle part of the sessions.

Parameter	Description
RT	Enable throttling based on the amount of time required for executing an SQL statement.
CPU	Enable throttling based on the CPU utilization. This option is not supported.
IO	Enable throttling based on the number of I/O operations. This option is not supported.
NETWORK	Enable throttling based on the network traffic during data transmission. This option is not supported.
QUEUE_TIME	Enable throttling based on the wait time for a query in the queue.
LOGICAL_READS	Enable throttling based on the number of logical reads. This option is not supported.
DISABLE SQL	Disable throttling.

Examples

- Assume that you want to throttle the sessions where the values for FOR PRIORITY are less than or equal to 100. Execute the following statement to throttle the sessions that wait longer than 0.1s in the queue:

```
alter system enable sql throttle for priority <= 100 using queue_time=0.1
```

UNIT

Description

You can use this statement to migrate units.

Syntax

```
alter_system_unit_stmt:
    ALTER SYSTEM MIGRATE
    UNIT [=] unit_id DESTINATION [=] ip_port

unit_id:
    INT_VALUE
```

Parameters

Parameter	Description
unit_id	The ID of the unit.
ip_port	The server IP address to which the unit is migrated.

Examples

- Migrate unit 1001 to 11.11.111.111:19510.

```
OceanBase(root@oceanbase)>alter system migrate unit = 1001 destination = '11.11.111.111:19510';
Query OK, 0 rows affected (0.05 sec)
```

ZONE

Description

You can use this statement to maintain zones. For example, you can add, delete, activate, or deactivate zones.

Syntax

```
alter_system_zone_stmt:
      ADD ZONE zone_name
      [zone_option_list]
    | {ALTER | CHANGE | MODIFY} ZONE zone_name
      [SET] zone_option_list
    | {DELETE | START | STOP | FORCE STOP} ZONE zone_name

zone_option_list:
    zone_option [, zone_option ...]

zone_option:
      region
    | idc
    | ZONE_TYPE {READONLY | READWRITE}

idc:
    STR_VALUE
```

Parameters

Parameter	Description
ADD ZONE	Add a zone.
{ALTER | CHANGE | MODIFY} ZONE	Modify the region property for a zone.
DELETE ZONE	Delete a zone. Before you delete a zone, ensure that no servers are available in the zone.

Parameter	Description
START | STOP	Activate or deactivate a zone.

Examples

- Delete a zone.

```
OceanBase(root@oceanbase)>alter system delete zone 'z1';
ERROR 4668 (HY000): The zone is not empty and can not be deleted. You should delete the servers of th
e zone. There are 1 servers alive and 0 not alive.
```

CLUSTER

Description

You can execute this statement to manage clusters. For example, you can add or delete a cluster, or modify cluster attributes.

Syntax

```
ALTER SYSTEM  cluster_action cluster_name CLUSTER_ID INTNUM;
cluster_action:
    ADD CLUSTER
    | REMOVE CLUSTER
    | ENABLE CLUSTER SYNCHRONIZATION
    | DISABLE CLUSTER SYNCHRONIZATION
```

Parameters

Parameter	Description
ADD CLUSTER	Add a secondary database.
REMOVE CLUSTER	Delete an existing secondary database.
ENABLE CLUSTER SYNCHRONIZATION	Enable data synchronization for a secondary database.
DISABLE CLUSTER SYNCHRONIZATION	Disable data synchronization for a secondary database.

Examples

- Add a secondary database.

```
ALTER SYSTEM ADD CLUSTER 'ob1.test' cluster_id = 1;
```

SWITCHOVER

Description

You can execute this statement to switch cluster roles. For example, you can switch a primary cluster to the secondary role, or switch a secondary cluster to the primary role.

Syntax

```
ALTER SYSTEM commit_switchover_clause;

commit_switchover_clause:
    COMMIT TO SWITCHOVER TO PRIMARY
    | COMMIT TO SWITCHOVER TO PHYSICAL STANDBY
    | ACTIVATE PHYSICAL STANDBY CLUSTER
    | CONVERT TO PHYSICAL STANDBY
```

Parameters

Parameter	Description
COMMIT TO SWITCHOVER TO PRIMARY	Switch a secondary cluster to the primary role. Execute this statement on a secondary cluster. Before you execute this statement, ensure that the original primary cluster is switched to the secondary role, and no another primary cluster exists.
COMMIT TO SWITCHOVER TO PHYSICAL STANDBY	Switch a primary cluster to the secondary role. Execute this statement on the primary cluster. Before you execute this statement, ensure that at least a secondary cluster is synchronized with the primary cluster. After you switch a primary cluster to the secondary role, you can also switch the cluster to the primary role again.
ACTIVATE PHYSICAL STANDBY CLUSTER	Switch a secondary cluster to the primary role if the primary cluster becomes faulty.
CONVERT TO PHYSICAL STANDBY	Switch a primary database to the secondary role.

Examples

- Switch a primary database to the secondary role.

```
ALTER SYSTEM COMMIT TO SWITCHOVER TO PHYSICAL STANDBY;
```

- Switch a secondary database to the primary role.

```
ALTER SYSTEM COMMIT TO SWITCHOVER TO PRIMARY;
```

- The primary cluster becomes faulty. Switch a secondary cluster to the primary role.

```
ALTER SYSTEM ACTIVATE PHYSICAL STANDBY CLUSTER;
```

- Restart the old primary database and switch it to the secondary role.

```
ALTER SYSTEM CONVERT TO PHYSICAL STANDBY;
```

BALANCE TASK

Description

You can use this statement to clear load balancing tasks that are not being scheduled.

Syntax

```
ALTER SYSTEM REMOVE BALANCE TASK opt_tenant_list opt_zone_list opt_balance_task_type;

opt_tenant_list
    TENANT [=] name,name_list

opt_zone_list
    ZONE [=] zone_name, zone_list

opt_balance_task_type
    ALL
    | MANUAL
    | AUTO
```

Parameters

Parameter	Description
opt_tenant_list	The tenants. If you leave this parameter empty, the tasks under all tenants are cleared.
opt_zone_list	The zones. This parameter is optional.
opt_balance_task_type	The type of the tasks to be cleared. Valid values: • ALL: all tasks. • AUTO: tasks that are automatically generated. • MANUAL: tasks that are manually generated.

Examples

- Clear the tasks that are not being scheduled under all tenants.

```
ALTER SYSTEM REMOVE BALANCE TASK;
```

CANCEL MIGRATE UNIT

Description

You can use this statement to cancel the migration of units.

Syntax

```
ALTER SYSTEM CANCEL MIGRATE UNIT unit_id;
```

Parameters

Parameter	Description
unit_id	The ID of the unit that is being migrated.

Examples

- Cancel the migration of unit 1001.

```
ALTER SYSTEM CANCEL MIGRATE UNIT 1001;
```

RESTORE

Description

You can use this statement to restore tenant data.

Syntax

```
alter system restore dest_tenant from source_tenant at 'uri' until 'timestamp' with 'restore_option';
```

Parameters

Parameter	Description
dest_tenant	The name of the new tenant to which the data is restored.
source_tenant	The name of the original tenant.
uri	The path from which the tenant data is restored.
timestamp	The timestamp to which the tenant data is restored. The value must be greater than or equal to START_TIME of baseline backup data in CDB_OB_BACKUP_SET_DETAILS, and less than or equal to MAX_NEXT_TIME of backup logs in CDB_OB_BACKUP_ARCHIVELOG_SUMMARY.

Parameter	Description
restore_option	The options for restoration. Valid values: • backup_cluster_name: the name of the source cluster. This parameter is required. • backup_cluster_id: the ID of the source cluster. This parameter is required. • pool_list: the resource pool of the user. This parameter is required. • locality: the locality of the tenant. This parameter is optional. • kms_encrypt: This parameter is optional. If you set kms_encrypt to true, specify kms_encrypt_info during restoration.

Examples

- Restore the tenant data.

```
alter system restore restored_trade from trade
    at 'oss://antsys-oceanbasebackup/backup_rd/20200323? host=cn-hangzhou-alipay-b.oss-cdn.aliyun-inc
.com&access_id=xxx&access_key=xxx'
  until ' 2020-03-23 08:59:45'
  with 'backup_cluster_name=ob20daily.backup&backup_cluster_id=1&pool_list=restore_pool';

alter system restore restored_trade from trade
  at 'file:///data/nfs/physical_backup_test/20200520'
  until '2020-05-21 09:39:54.071670'
  with 'backup_cluster_name=ob20daily.backup&backup_cluster_id=1&pool_list=restore_pool&locality=F@z1
,F@z2,F@z3';
```

CHANGE TENANT

Description

You can use this statement to switch to another tenant.

Syntax

```
ALTER SYSTEM CHANGE TENANT tenant_name;

ALTER SYSTEM CHANGE TENANT TENANT_ID [=] INTNUM;
```

Parameters

Parameter	Description
tenant_name	The name of the tenant that you want to switch to.
TENANT_ID	The ID of the tenant that you want to switch to.

Examples

- Switch to the tenant whose ID is 1001.

```
ALTER SYSTEM CHANGE TENANT TENANT_ID = 1001;
```

Notes

1. Log on to the database as a system tenant. A general tenant cannot execute this statement.
2. Execute this statement on an OBServer that is running properly. If an OBServer is disconnected, execute this statement again after the OBServer is recovered.
3. You cannot switch to another tenant when transactions are being processed.
4. If you switch to a non-system tenant, you cannot execute data definition language (DDL) statements.

BACKUP

Description

You can use this statement to trigger data backup.

Syntax

```
Specify a path to store the backup data: alter system set backup_dest = <backup_uri>
Enable log archiving: alter system archivelog
Disable log archiving: alter system noarchivelog
Back up baseline data for a cluster: alter system backup database;
Cancel the current backup task: alter system cancel backup
```

Parameters

Parameter	Description
backup_uri	The path to store the backup data. You can specify an Object Storage Service (OSS) path or a file system. For more information about the formats, see Examples.

Examples

- Set the path to store the backup data.

```
alter system set backup_dest='oss://antsys-oceanbasebackup/backup_dir? host=xxx&access_id=xxx&access_
key=xxx';
alter system set backup_dest='file:///data/nfs/physical_backup_dir';
```

Notes

Log on to the database as a system tenant. A general tenant cannot execute this statement.

17.1.4.5.7. ALTER TABLE

Description

You can execute the ALTER TABLE statement to update the schema of an existing table. For example, you can modify an existing table and table attributes, add columns, modify columns and column attributes, or delete columns.

Syntax

```
alter_table_stmt:
      ALTER TABLE table_name
      alter_table_action_list;
    | RENAME TABLE rename_table_action_list;

alter_table_action_list:
    alter_table_action [, alter_table_action ...]

alter_table_action:
      ADD [COLUMN] {column_definition | (column_definition_list)}
    | CHANGE [COLUMN] column_name column_definition
    | MODIFY [COLUMN] column_definition
    | ALTER [COLUMN] column_name {SET DEFAULT const_value | DROP DEFAULT}
    | DROP [COLUMN] column_name
    | ADD [CONSTRAINT [constraint_name]] UNIQUE {INDEX | KEY} [index_name] index_desc
    | ADD {INDEX | KEY} [index_name] index_desc
    | ADD FULLTEXT [INDEX | KEY] [index_name] fulltext_index_desc
    | ALTER INDEX index_name [VISIBLE | INVISIBLE]
    | DROP {INDEX | KEY} index_name
    | ADD PARTITION (range_partition_list)
    | DROP PARTITION (partition_name_list)
    | REORGANIZE PARTITION name_list INTO partition_range_or_list
    | TRUNCATE PARTITION name_list
    | [SET] table_option_list
    | RENAME [TO] table_name
    | DROP TABLEGROUP
    | DROP FOREIGN KEY fk_name

rename_table_action_list:
    rename_table_action [, rename_table_action ...]

rename_table_action:
    table_name TO table_name

column_definition_list:
    column_definition [, column_definition ...]

column_definition:
    column_name data_type
    [DEFAULT const_value] [AUTO_INCREMENT]
    [NULL | NOT NULL] [[PRIMARY] KEY] [UNIQUE [KEY]] comment

index_desc:
   (column_desc_list) [index_type] [index_option_list]

fulltext_index_desc:
   (column_desc_list) CTXCAT(column_desc_list) [index_option_list]

column_desc_list:
    column_desc [, column_desc ...]

column_desc:
     column_name [(length)] [ASC | DESC]

index_type:
    USING BTREE
```

```
index_option_list:
    index_option [ index_option ...]

index_option:
      [GLOBAL | LOCAL]
    | block_size
    | compression
    | STORING(column_name_list)
    | comment

table_option_list:
    table_option [ table_option ...]

table_option:
    | primary_zone
    | replica_num
    | table_tablegroup
    | block_size
    | compression
    | AUTO_INCREMENT [=] INT_VALUE
    | comment
    | DUPLICATE_SCOPE [=] "none|zone|region|cluster"

partition_option:
      PARTITION BY HASH(expression)
      [subpartition_option] PARTITIONS partition_count
    | PARTITION BY KEY([column_name_list])
      [subpartition_option] PARTITIONS partition_count
    | PARTITION BY RANGE {(expression) | COLUMNS (column_name_list)}
      [subpartition_option] (range_partition_list)

subpartition_option:
      SUBPARTITION BY HASH(expression)
      SUBPARTITIONS subpartition_count
    | SUBPARTITION BY KEY(column_name_list)
      SUBPARTITIONS subpartition_count
    | SUBPARTITION BY RANGE {(expression) | COLUMNS (column_name_list)}
      (range_subpartition_list)

range_partition_list:
    range_partition [, range_partition ...]

range_partition:
    PARTITION partition_name
    VALUES LESS THAN {(expression_list) | MAXVALUE}

range_subpartition_list:
    range_subpartition [, range_subpartition ...]

range_subpartition:
    SUBPARTITION subpartition_name
    VALUES LESS THAN {(expression_list) | MAXVALUE}

expression_list:
    expression [, expression ...]

column_name_list:
    column_name [, column_name ...]

partition_name_list:
```

```
    partition_name [, partition_name ...]

partition_count | subpartition_count:
    INT_VALUE
```

Parameters

Parameter	Description
ADD [COLUMN]	Add the column. You cannot add a primary key column.
CHANGE [COLUMN]	Modify the column name and column attributes.
MODIFY [COLUMN]	Modify column attributes.
ALTER [COLUMN]	Modify the default value of the column.
DROP [COLUMN]	Delete the column. You cannot delete a primary key column or column that has indexes.
ADD [UNIQUE INDEX]	Add the unique index.
ADD [INDEX]	Add the common index.
ALTER [INDEX]	Modify index attributes.
ADD [PARTITION]	Add the partition.
DROP [PARTITION]	Delete the partition.
REORGANIZE [PARTITION]	Re-partition the table.
TRUNCATE [PARTITION]	Delete data from the partition.
RENAME [TO] table_name	Rename the table.
DROP [TABLEGROUP]	Delete the table group.
DROP [FOREIGN KEY]	Delete the foreign key.
SET BLOCK_SIZE	The block size for the partitioned table.

Parameter	Description
SET REPLICA_NUM	The number of replicas for the table.
SET COMPRESSION	The compression method for the table.
SET USE_BLOOM_FILTER	Specifies whether to use a Bloom filter.
SET COMMENT	Add comments.
SET PROGRESSIVE_MERGE_NUM	The number of macro-blocks that are to be merged at a time for progressive compaction. Valid values: 1 to 64.

Examples

- Change the field name d in the t2 table to c and change the data type of the field.

```
ALTER TABLE t2 CHANGE COLUMN d c CHAR(10);
```

- Add a column, and then delete it.
  - Before you add a column, execute the `DESCRIBE test;` statement to view the table. The following information appears:

```
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| c1    | int(11)     | NO   | PRI | NULL    |       |
| c2    | varchar(50) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

  - Execute the following statement to add the c3 column:

```
ALTER TABLE test ADD c3 int;
```

  - After you add the column, execute the `DESCRIBE test;` statement to view the table. The following information appears:

```
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| c1    | int(11)     | NO   | PRI | NULL    |       |
| c2    | varchar(50) | YES  |     | NULL    |       |
| c3    | int(11)     | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.02 sec)
```

  - Execute the following statement to delete the c3 column:

```
ALTER TABLE test DROP c3;
```

- After you delete the column, execute the `DESCRIBE test;` statement to view the table. The following information appears:

```
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| c1    | int(11)     | NO   | PRI | NULL    |       |
| c2    | varchar(50) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

- Configure the number of replicas for the test table, and add the c5 column to the table.

```
ALTER TABLE test SET REPLICA_NUM=2, ADD COLUMN c5 INT;
```

17.1.4.5.8. ALTER TABLEGROUP

Description

You can execute the ALTER TABLEGROUP statement to perform the following operations:

- Add multiple tables to a table group.
- Modify partition rules for a table group.
- Modify the locality and primary zone for a table group.

Syntax

- Add multiple tables to a table group.

```
ALTER TABLEGROUP tablegroupname ADD [TABLE] tblname [, tblname…]
```

- Modify partition rules for a table group.

```
ALTER TABLEGROUP tablegroupname alter_tg_partition_option
alter_tg_partition_option:
DROP PARTITION '(' name_list ')'
| ADD PARTITION opt_range_partition_list
| modify_tg_partition_info
```

- Modify the locality and primary zone for a table group.

```
ALTER TABLEGROUP tablegroupname alter_tablegroup_actions
alter_tablegroup_actions:
alter_tablegroup_action
| alter_tablegroup_action, alter_tablegroup_action
alter_tablegroup_action:
SET LOCALITY [=] locality_name
|SET PRIMARY_ZONE [=] primary_zone_name
```

Parameters

Parameter	Description
tablegroupname	The name of the table group.

Parameter	Description
tblname	The name of the table. To add multiple tables to a table group, separate table names with commas (,). If you add multiple tables at a time, you can specify duplicate table names. If the table you want to add exists in the *specified table group*, the system does not return an error.
modify_tg_partition_info	Modify the partition rules for the table group.
LOCALITY locality_name	The locality of the table group.
PRIMARY_ZONE primary_zone_name	The primary zone of the table group.

Examples

Create table group tgh and two relational tables ttgh and ttgh2, and change the locality of the table group to F@z1.

```
OceanBase(admin@test)> create tablegroup tgh locality='F,R{ALL_SERVER}@z1' partition by hash partitio
ns 10;
Query OK, 0 rows affected (0.09 sec)

OceanBase(admin@test)> create table ttgh(c1 int, c2 int) tablegroup = tgh locality='F,R{ALL_SERVER}@z
1';
Query OK, 0 rows affected (0.55 sec)

OceanBase(admin@test)> create table ttgh2(c1 int, c2 int) tablegroup = tgh locality='F,R{ALL_SERVER}@
z1';
Query OK, 0 rows affected (0.39 sec)

OceanBase(admin@test)> alter tablegroup tgh set locality ='F@z1';
Query OK, 0 rows affected (0.09 sec)

OceanBase(admin@test)> select locality from oceanbase.__all_tablegroup where tablegroup_name ='tgh';
+------------+
| locality   |
+------------+
| FULL{1}@z1 |
+------------+
1 row in set (0.05 sec)

OceanBase(admin@test)> select locality from oceanbase.__all_table where tablegroup_id=(select tablegr
oup_id from oceanbase.__all_tablegroup where tablegroup_name ='tgh');
+------------+
| locality   |
+------------+
| FULL{1}@z1 |
| FULL{1}@z1 |
+------------+
2 rows in set (0.04 sec)
```

17.1.4.5.9. ALTER TENANT

Description

You can execute the ALTER TENANT statement to modify tenant information.

Syntax

```
ALTER TENANT {tenant_name | ALL}
    [SET] [tenant_option_list] [opt_global_sys_vars_set]

tenant_option_list:
    tenant_option [, tenant_option ...]

tenant_option:
            COMMENT [=]'string'
            |{CHARACTER SET | CHARSET} [=] charsetname
            |COLLATE [=]  collationname
            |REPLICA_NUM [=] num
            |ZONE_LIST [=] (zone [, zone…])
            |PRIMARY_ZONE [=] zone
            |RESOURCE_POOL_LIST [=](poolname [, poolname…])
            |DEFAULT TABLEGROUP [=] {NULL | tablegroupname}
            |{READ ONLY | READ WRITE}
      |LOGONLY_REPLICA_NUM [=] num
      |LOCALITY [=] 'locality description'
      |LOCK|UNLOCK;

opt_global_sys_vars_set:
      VARIABLES system_var_name = expr [,system_var_name = expr] ...
```

Parameters

Parameter	Description
tenant_name	The name of the tenant that you want to modify.
ALTER TENANT ALL	Modify information for all tenants at a time.
RESOURCE_POOL_LIST	The resource pools. This parameter is required when you create a tenant. You can specify only one resource pool.
DEFAULT TABLEGROUP	The default table group of the tenant. A NULL value specifies no default table group for the database.
LOCK|UNLOCK	Lock or unlock the tenant. After the tenant is locked, you cannot create sessions on the tenant. The existing sessions remain unchanged. You can lock a tenant if the subscription of the service is not renewed upon expiration. After the subscription is renewed, you can unlock the tenant.

Parameter	Description
COMMENT	The comments.
CHARACTER SET | CHARSET	The character set for the tenant.
COLLATE	The collation.
REPLICA_NUM	The number of replicas.
ZONE_LIST	The zones.
PRIMARY_ZONE	The primary zone.
READ ONLY | READ WRITE	The attribute of the tenant. Valid values: READ ONLY and READ WRITE.
LOGONLY_REPLICA_NUM	The number of log replicas.
LOCALITY	The distribution of replicas across zones. For example, the F@z1,F@z2,F@z3,R@z4 value specifies that replicas in the z1, z2, and z3 zones are FULL replicas, and replicas in the z4 zone are READONLY replicas.
system_var_name	The system variable for the tenant.

Examples

- Lock the TENANT1 tenant.

```
ALTER TENANT TENANT1 LOCK;
```

Notes

The system tenants and administrator of the current tenant have the permission to execute the ALTER TENANT statement.

17.1.4.5.10. ALTER USER

Description

You can execute the ALTER USER statement to perform the following operations:

- Change the password of a user that is used to log on to the ApsaraDB for OceanBase system.
- Lock or unlock a user. A locked user cannot log on to the ApsaraDB for OceanBase system.

> **Note**
>
> To execute the ALTER USER statement, ensure you have the global UPDATE USER permission.

Syntax

- Change the password of a user.

```
ALTER USER 'username' IDENTIFIED BY 'password';
```

- Lock a user.

```
ALTER USER user [lock_option]

lock_option:{
ACCOUNT LOCK
| ACCOUNT UNLOCK}
```

Parameters

Parameter	Description
ACCOUNT UNLOCK	Lock the user.
ACCOUNT UNLOCK	Unlock the user.

Examples

- Change a password.

Execute the following statement to change the password of the sqluser01 user to abc123:

```
ALTER USER 'sqluser01' IDENTIFIED BY 'abc123';
```

- Lock a user.

Lock the obsqluser01 user.

```
ALTER USER 'obsqluser01' ACCOUNT LOCK;
```

- Unlock a user.

Unlock the obsqluser01 user.

```
ALTER USER 'obsqluser01' ACCOUNT UNLOCK;
```

17.1.4.5.11. CREATE DATABASE

Description

You can execute the CREATE DATABASE statement to create a database. You can also specify default attributes for a database, such as the default character set and collation of the database.

Syntax

```
create_database_stmt:
    CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] database_name [database_option_list]

database_option_list:
    database_option [database_option ...]

database_option:
    [DEFAULT] {CHARACTER SET | CHARSET} [=] charset_name
  | [DEFAULT] COLLATE [=] collation_name
  | REPLICA_NUM [=] int_num
  | PRIMARY_ZONE [=] zone_name
  | {READ ONLY | READ WRITE}
  | DEFAULT TABLEGROUP [=] {NULL | table_group_name}
```

Parameters

Parameter	Description
database_name	The name of the database for which you want to modify attributes. By default, the current database is modified if you leave this parameter empty.
CHARSET charset_name	The character set that you want to modify.
COLLATE collation_name	The collation.
REPLICA_NUM int_num	The number of replicas.
PRIMARY_ZONE zone_name	The primary zone.
READ ONLY | READ WRITE	The attribute of the database. Valid values: READ ONLY and READ WRITE.
DEFAULT TABLEGROUP table_group_name	The default table group of the database. A NULL value specifies no default table group for the database.

Examples

- Create the test2 database that uses the UTF-8 character set.

```
OceanBase(admin@test)>create database test2 default CHARACTER SET UTF8;
Query OK, 1 row affected (0.00 sec)
```

- Create the test3 database in read/write mode.

```
OceanBase(admin@test)>create database test3 READ WRITE;
Query OK, 1 row affected (0.03 sec)
```

17.1.4.5.12. CREATE INDEX

Description

You can execute the CREATE INDEX statement to create an index. Indexes are created on tables to sort the values of one or more table columns. Indexes are used to reduce the query response time and performance overhead of database systems.

Syntax

```
CREATE [UNIQUE] INDEX indexname
         ON tblname (index_col_name,...)
      [index_type] [index_options]
index_type:
      USING BTREE

index_options:
      index_option [index_option…]

index_option:
    GLOBAL | LOCAL
    | COMMENT 'string'
    | COMPRESSION [=] {NONE | LZ4_1.0 | LZO_1.0 | SNAPPY_1.0 | ZLIB_1.0}
    | BLOCK_SIZE [=] size
    | STORING(columname_list)
    | VISIBLE | INVISIBLE

index_col_name:
    colname [(length)] [ASC | DESC]

columname_list:
    colname [, colname…]
```

Parameters

Parameter	Description
indexname	The name of the index that you want to create.
tblname	The name of the table to which the index belongs.
index_col_name	The name of the column on which the index is created. You can specify ASC that follows each column name. ASC indicates that the values are sorted in ascending order. DESC is not supported. By default, the values are sorted in ascending order. In the CREATE INDEX statement, the values of the first column in index_col_name are indexed. If the values in the first column are the same, the values in the next column are indexed. Similar rules apply to the other columns.
index_type	The type of the index. Set the value to USING BTREE.

Parameter	Description
UNIQUE	The unique index.
index_option	The index option. To specify multiple index options, separate them with spaces.
GLOBAL | LOCAL	Specifies whether the index is a global or local index. Default value: GLOBAL.
COMMENT	The comments.
COMPRESSION	The compression algorithm.
BLOCK_SIZE	The size of a micro-block.
STORING	Some columns are stored in the indexed table for redundant storage. This improves the system query performance.

Examples

1. Execute the following statement to create the test table:

```
CREATE TABLE test (c1 int primary key, c2 VARCHAR(10));
```

2. Execute the following statement to create indexes on the test table:

```
CREATE INDEX test_index ON test (c1, c2 DESC);
```

3. Execute the following statement to view the indexes of the test table:

```
SHOW INDEX FROM test;
```

17.1.4.5.13. CREATE OUTLINE

Description

You can execute the CREATE OUTLINE statement to create an outline. You can use an SQL text or SQL statement ID to create an outline. An SQL text indicates an original SQL statement that contains parameters.

> **Note**
>
> Before you create an outline, log on to the database for which you want to create the outline.

Syntax

- **Use an SQL text to create an outline.**

```
CREATE [OR REPLACE] OUTLINE outline_name ON stmt [ TO target_stmt ]
```

- **Use an SQL statement ID to create an outline.**

```
CREATE OUTLINE outline_name ON sql_id USING HINT  hint;
```

Parameters

Parameter	Description
outline_name	The name of the outline that you want to create.
OR REPLACE	Replace an existing outline with a new outline of the same name.
stmt	The original data manipulation language (DML) statement that contains hints and parameters.
TO target_stmt	You can leave TO target_stmt empty if the following requirements are met: The original parameterized SQL statement is equivalent to the stmt value without hints. In this case, the system generates an execution plan for the parameterized SQL statement that has the hints in stmt. To generate a fixed execution plan for a statement that contains hints, specify the original SQL statement as TO target_stmt. **Notice** Make sure TO target_stmt you specify is the same as the stmt value without hints.
sql_id	Assume that the SQL statement of the specified ID contains hints. When you create an outline, the hints you specify overwrite all existing hints in the original statement.
hint	The value must be in the /*+ xxx */ format.

Examples

- Use an SQL text to create an outline.

```
CREATE OUTLINE otl_idx_c2
       ON SELECT/*+ index(t1 idx_c2)*/ * FROM t1 WHERE c2 = 1;
```

- Use an SQL statement ID to create an outline.

```
CREATE OUTLINE otl_idx_c2
ON "ED570339F2C856BA96008A29EDF04C74"
USING HINT /*+ index(t1 idx_c2)*/ ;
```

Notes

The outline created by using an SQL text prevails and overwrites that created by using an SQL statement ID.

17.1.4.5.14. CREATE RESOURCE POOL

Description

You can execute the CREATE RESOURCE POOL statement to create a resource pool. A resource pool represents a set of resource units that can be assigned to tenants.

Syntax

```
CREATE RESOURCE POOL poolname
  UNIT [=] unitname,
  UNIT_NUM [=] unitnum,
  ZONE_LIST [=] ('zone' [, 'zone' …]) ;
```

Parameters

Parameter	Description
UNIT	The name of the resource unit that you want to create.
NIT_NUM	The number of resource units that you want to create in a zone. The value must be smaller than the number of OBServers in the zone.
ZONE_LIST	The cluster to which the resource pool that you want to create belongs.

Examples

- Create resource pool pool1 in cluster zone1.

```
OceanBase(admin@test)> CREATE RESOURCE POOL pool1 unit='unit1', unit_num=1, zone_list=('zone1');
Query OK, 0 rows affected (0.01 sec)
```

17.1.4.5.15. CREATE RESOURCE UNIT

Description

You can execute the CREATE RESOURCE UNIT statement to create a resource unit. In this statement, you can specify parameters to assign hardware resources to tenants.

Syntax

```
CREATE RESOURCE UNIT unitname
MAX_CPU [=] cpunum,
MAX_MEMORY [=] memsize,
MAX_IOPS [=] iopsnum,
MAX_DISK_SIZE [=] disksize,
MAX_SESSION_NUM [=] sessionnum,
[MIN_CPU [=] cpunum,]
[MIN_MEMORY [=] memsize,]
[MIN_IOPS [=] iopsnum] ;
```

Parameters

Parameter	Description
MAX_CPU	The maximum number of CPUs.
MAX_MEMORY	The maximum memory size. Valid values: [1073741824,+∞). The unit is bytes. The minimum value 1073741824 bytes is equivalent to 1 GB.
MAX_IOPS	The maximum IOPS. Valid values: [128,+∞).
MAX_DISK_SIZE	The maximum disk capacity. Valid values: [536870912,+∞). The unit is bytes. The minimum value 536870912 bytes is equivalent to 512 MB.
MAX_SESSION_NUM	The maximum number of sessions. Valid values: [64,+∞).
MIN_CPU	The minimum number of CPUs.
MIN_MEMORY	The minimum memory size.
MIN_IOPS	The minimum IOPS.

Examples

- Create resource unit unit1. The resource unit has one CPU, 1 GB memory, 128 IOPS, 10 GB disk capacity, and 64 sessions.

```
OceanBase(admin@test)> CREATE RESOURCE UNIT unit1 max_cpu 1, max_memory '1G', max_iops 128,max_disk_s
ize '10G', max_session_num 64, MIN_CPU=1, MIN_MEMORY= '1G', MIN_IOPS=128;
Query OK, 0 rows affected (0.02 sec)
```

- Create resource unit unit1. The resource unit has one CPU, 1 GB memory, 128 IOPS, 10 GB disk capacity, and 64 sessions.

```
OceanBase(admin@test)> CREATE RESOURCE UNIT unit1 max_cpu 1, max_memory 1073741824, max_iops 128, max
_disk_size 10737418240, max_session_num 64, MIN_CPU=1, MIN_MEMORY=1073741824, MIN_IOPS=128;
Query OK, 0 rows affected (0.01 sec)
```

17.1.4.5.16. CREATE SYNONYM

Description

You can execute the CREATE SYNONYM statement to create a synonym.

Syntax

```
CREATE [ OR REPLACE ] [ PUBLIC ]
SYNONYM [ DATABASE. ]synonym
FOR [ DATABASE. ]object;
```

Parameters

Parameter	Description
OR REPLACE	Create the synonym again if it already exists. You can use this clause to change the definition of an existing synonym.
PUBLIC	Add PUBLIC to the statement to create a public synonym. Public synonyms are accessible to all users. To use a public synonym, ensure you have appropriate permissions on the underlying object. The system uses a public synonym only if you do not specify the DataBase parameter for the object. If you do not use the PUBLIC keyword, the synonym is private and is accessible only in the current database. The name of a private synonym must be unique in the current database.
[DataBase.]synonym	DataBase specifies the database to which the current synonym belongs. If you add PUBLIC to the statement, you do not need to specify the database for the synonym. The synonym parameter specifies the name of the synonym.
[DataBase.]object	The name of the object for which the synonym is created.

Examples

- Create a synonym.

```
OceanBase(admin@test)>create table t1(c1 int);
Query OK, 0 rows affected (0.18 sec)

OceanBase(admin@test)>create synonym s1 for t1;
Query OK, 0 rows affected (0.05 sec)

OceanBase(admin@test)>insert into s1 values(1);
Query OK, 1 row affected (0.02 sec)

OceanBase(admin@test)>select * from s1;
+------+
| c1   |
+------+
|    1 |
+------+
1 row in set (0.01 sec)
```

- Create a public synonym.

```
OceanBase(admin@test)>create public synonym syn_pub for t1;
Query OK, 0 rows affected (0.03 sec)

OceanBase(admin@test)>select * from syn_pub;
+------+
| c1   |
+------+
|    1 |
+------+
1 row in set (0.01 sec)
```

Notes

To create a synonym, ensure that you have the following permissions:

- To create a private synonym for the current database, ensure that you have the CREATE SYNONYM permission.
- To create a private synonym for another database, ensure that you have the CREATE ANY SYNONYM permission.
- To create a public synonym, ensure that you have the CREATE PUBLIC SYNONYM permission.
- The object for which you want to create a synonym does not need to exist. Therefore, you do not need to have the permission to access the object.

```
Connect to the database as the syn_user user.
OceanBase(ADMIN@TEST)>CREATE USER syn_user IDENTIFIED BY syn_user;
Query OK, 0 rows affected (0.06 sec)

OceanBase(ADMIN@TEST)>grant CREATE on syn_user.* to syn_user;
Query OK, 0 rows affected (0.02 sec)

OceanBase(ADMIN@TEST)>grant SELECT on syn_user.* to syn_user;
Query OK, 0 rows affected (0.03 sec)


Connect to the database as the syn_user user.
//Failed to create a synonym.
OceanBase(SYN_USER@(none))>create synonym syn_1 for t1;
ERROR-00600: internal error code, arguments: -5036, Access denied; you need (at least one of) the CRE
ATE SYNONYM privilege(s) for this operation

Connect to the database as the syn_user user and grant the CREATE SYNONYM permission to the user.
OceanBase(ADMIN@TEST)>grant CREATE SYNONYM on *.* to syn_user;
Query OK, 0 rows affected (0.03 sec)

Connect to the database as the syn_user user.
OceanBase(SYN_USER@(none))>create synonym syn_1 for t1;
Query OK, 0 rows affected (0.05 sec)
```

17.1.4.5.17. CREATE TABLE

Description

You can execute the CREATE TABLE statement to create a table in a database.

Syntax

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] table_name
      (table_definition_list) [table_option_list] [partition_option] [AS] select;

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] table_name
      LIKE table_name;

table_definition_list:
    table_definition [, table_definition ...]

table_definition:
      column_definition
    | [CONSTRAINT [constraint_name]] PRIMARY KEY index_desc
    | [CONSTRAINT [constraint_name]] UNIQUE {INDEX | KEY} [index_name] index_desc
    | {INDEX | KEY} [index_name] index_desc
    | FULLTEXT [INDEX | KEY] [index_name] fulltext_index_desc

column_definition_list:
    column_definition [, column_definition ...]

column_definition:
    column_name data_type
    [DEFAULT const_value] [AUTO_INCREMENT]
    [NULL | NOT NULL] [[PRIMARY] KEY] [UNIQUE [KEY]] comment

index_desc:
```

```
   (column_desc_list) [index_type] [index_option_list]

fulltext_index_desc:
   (column_desc_list) CTXCAT(column_desc_list) [index_option_list]

column_desc_list:
    column_desc [, column_desc ...]

column_desc:
     column_name [(length)] [ASC | DESC]

index_type:
    USING BTREE

index_option_list:
    index_option [ index_option ...]

index_option:
      [GLOBAL | LOCAL]
    | block_size
    | compression
    | STORING(column_name_list)
    | comment

table_option_list:
    table_option [ table_option ...]

table_option:
      [DEFAULT] {CHARSET | CHARACTER SET} [=] charset_name
    | [DEFAULT] COLLATE [=] collation_name
    | primary_zone
    | replica_num
    | table_tablegroup
    | block_size
    | compression
    | AUTO_INCREMENT [=] INT_VALUE
    | comment
    | DUPLICATE_SCOPE [=] "none|zone|region|cluster"
    | LOCALITY [=] "locality description"
    | ROW_FORMAT [=] REDUNDANT|COMPACT|DYNAMIC|COMPRESSED|DEFAULT
    | PCTFREE [=] num

partition_option:
      PARTITION BY HASH(expression)
      [subpartition_option] PARTITIONS partition_count
    | PARTITION BY KEY([column_name_list])
      [subpartition_option] PARTITIONS partition_count
    | PARTITION BY RANGE {(expression) | COLUMNS (column_name_list)}
      [subpartition_option] (range_partition_list)
    | PARTITION BY LIST {(expression) | COLUMNS (column_name_list)}
      [subpartition_option] PARTITIONS partition_count

subpartition_option:
      SUBPARTITION BY HASH(expression)
      SUBPARTITIONS subpartition_count
    | SUBPARTITION BY KEY(column_name_list)
      SUBPARTITIONS subpartition_count
    | SUBPARTITION BY RANGE {(expression) | COLUMNS (column_name_list)}
      (range_subpartition_list)
    | SUBPARTITION BY LIST(expression)
```

```
    | SUBPARTITION BY LIST(expression)

range_partition_list:
    range_partition [, range_partition ...]

range_partition:
    PARTITION partition_name
    VALUES LESS THAN {(expression_list) | MAXVALUE}

range_subpartition_list:
    range_subpartition [, range_subpartition ...]

range_subpartition:
    SUBPARTITION subpartition_name
    VALUES LESS THAN {(expression_list) | MAXVALUE}

expression_list:
    expression [, expression ...]

column_name_list:
    column_name [, column_name ...]

partition_name_list:
    partition_name [, partition_name ...]

partition_count | subpartition_count:
    INT_VALUE
```

Parameters

Parameter	Description
DUPLICATE_SCOPE	The attribute of the replicated table. Valid values: • none: The table is a standard table. • zone: The table is a replicated table. The replica leader must replicate transactions to all FULL (F) replicas and READONLY (R) replicas in the current zone. • region: The table is a replicated table. The replica leader must replicate transactions to all F replicas and R replicas in the current region. • cluster: The table is a replicated table. The replica leader must replicate transactions to all F replicas and R replicas in the current cluster. If you do not specify DUPLICATE_SCOPE, the default value is none.

Parameter	Description

ROW_FORMAT	Specifies whether to enable data encoding for the table.
	• redundant
	  ○ Disable data encoding.
	• compact
	  ○ Disable data encoding.
	• dynamic
	  ○ Enable data encoding.
	• compressed
	  ○ Enable data encoding.
	• default
	  ○ If you set the value to default, the dynamic mode is used.

BLOCK_SIZE	The micro-block size of the table.
COMPRESSION	The compression algorithm used by the table. Valid values: 1. none: No compression algorithms are used. 2. lz4_1.0: The lz4 compression algorithm is used. 3. zstd_1.0: The zstd compression algorithm is used. 4. snappy_1.0: The snappy compression algorithm is used.
CHARSET | CHARACTER SET	The default character set of the columns in the table. Valid values: utf8, utf8mb4, gbk, utf16, and gb18030.
COLLATE	The default comparison rules of the columns in the table. Valid values: utf8_bin, utf8_general_ci, utf8_unicode_ci, gbk_bin, gbk_chinese_ci, utf8mb4_general_ci, utf8mb4__general_cs, utf8mb4_bin, utf8mb4_unicode_ci, utf16_general_ci, utf16_bin, utf16_unicode_ci, gb18030_chinese_ci, gb18030_bin
primary_zone	The primary zone where the replica leader resides.
replica_num	The number of replicas.
table_tablegroup	The table group to which the table belongs.

Parameter	Description
AUTO_INCREMENT	The initial value of the auto-increment column in the table.
comment	The comments.
LOCALITY	A description about the distribution of replicas across zones. For example, the F@z1,F@z2,F@z3,R@z4 value specifies that replicas in the z1, z2, and z3 zones are F replicas and the replicas in the z4 zone are R replicas.
PCTFREE	The percentage of the idle space reserved in a macro-block.

Examples

- Create a table in a database.

```
CREATE TABLE test (c1 int primary key, c2 VARCHAR(50)) REPLICA_NUM = 3, PRIMARY_ZONE = 'zone1';
```

- Create a table and enable vertical partitioning for the table. The first partition contains only the c3 column. The second partition contains the c1 and c2 columns. The third partition contains the c4 and c5 columns that are not listed.

```
CREATE TABLE t1(c1 int,
                c2 int,
                c3 int,
                c4 int,
                c5 int)
                PARTITION BY CLOUMN ( c3, (c1, c2));
```

- Create a replicated table.

```
CREATE TABLE item() locality = 'F,R{all_server}@hz1, F,R{all_server}@hz2,
F,R{all_server}@hz3' DUPLICATE_SCOPE="cluster"
```

- Create a table that has an index.

```
create table t1 (c1 int primary key, c2 int, c3 int, index i1 (c2));
```

- Create an eight-partition table that uses hash partitioning.

```
create table t1 (c1 int primary key, c2 int) partition by hash(c1) partitions 8;
```

- Create a table that uses range partitioning to determine the partitions and uses key partitioning to determine the subpartitions.

```
 create table t1 (c1 int, c2 int, c3 int)
     partition by range(c1) subpartition by key(c2, c3) subpartitions 5
     (partition p0 values less than(0), partition p1 values less than(100));
```

- Create a table in which one column uses the GBK character set and the other column uses the UTF-8 character set.

```
create table t1 (c1 varchar(10),
                 c2 varchar(10) charset gbk collate gbk_bin)
  default charset utf8 collate utf8mb4_general_ci;
```

- Enable encoding, use the zstd compression algorithm, and reserve five percent of the space in a macro-block.

```
create table t1 (c1 int, c2 int, c3 varchar(64))
  compression 'zstd_1.0'
  ROW_FORMAT dynamic
  pctfree 5;
```

17.1.4.5.18. CREATE TABLEGROUP

Description

You can execute the CREATE TABLEGROUP statement to create a table group.

> **Note**
>
> Only an administrator under a tenant can create a table group.

Syntax

```
CREATE TABLEGROUP [IF NOT EXISTS] tablegroupname [opt_tablegroup_option_list] [opt_tg_partition_optio
n]

opt_tablegroup_option_list:
tablegroup_option [tablegroup_option]

tablegroup_option:
LOCALITY [=] locality_name
| PRIMARY_ZONE [=] primary_zone_name

opt_tg_partition_option:
PARTITION BY
 KEY COLUMN_NUM [tg_subpartition_option] PARTITIONS INTNUM
| HASH [tg_subpartition_option] PARTITIONS INTNUM
| RANGE [tg_subpartition_option] {PARTITION partition_name VALUES LESS THAN range_partition_expr, ...
}
| RANGE COLUMNS COLUMN_NUM [tg_subpartition_option] {PARTITION partition_name VALUES LESS THAN range_
partition_expr, ...}
| LIST [tg_subpartition_option] {PARTITION partition_name VALUES in list_partition_expr, ...}
| LIST COLUMNS COLUMN_NUM [tg_subpartition_option] {PARTITION partition_name VALUES in list_partition
_expr, ...}

tg_subpartition_option:
SUBPARTITION BY
 RANGE SUBPARTITION TEMPLATE {SUBPARTITION partition_name VALUES LESS THAN range_partition_expr, ...}
| RANGE COLUMNS COLUMN_NUM SUBPARTITION TEMPLATE {SUBPARTITION partition_name VALUES LESS THAN range_
partition_expr, ...}
| HASH [SUBPARTITIONS INTNUM]
| KEY COLUMN_NUM [SUBPARTITIONS INTNUM]
| LIST SUBPARTITION TEMPLATE {SUBPARTITION partition_name VALUES in list_partition_expr, ...}
| LIST COLUMNS COLUMN_NUM SUBPARTITION TEMPLATE {SUBPARTITION partition_name VALUES in list_partition
_expr, ...}
```

Parameters

Parameter	Description
tablegroupname	A table group name must be up to 64 characters in length and can contain letters, digits, and underscores (_). The name must start with a letter or an underscore (_) and cannot be a keyword that is reserved for ApsaraDB for OceanBase. If the specified tenant name is used and the IF NOT EXISTS option is not specified, the system returns an error.

Parameter	Description
opt_tablegroup_option_list	The partition mode, locality, and primary zone of a table group must be the same as those of the tables in the table group. You cannot modify a single table record in a table group. You can only modify a table group to modify multiple table records at a time. Tables in a table group must be in the same locality. The type, number, and locality of replicas must be the same. Tables in a table group must be in the same primary zone and have the same replica leader. The priority of the replicas is the same. Tables in a table group are partitioned in the same way. • Specify the same partitioning type for all tables. For example, partition all tables in hash-range partitioning mode. • To use key partitioning, reference the same number of columns in each table, and ensure that each table has the same number of partitions. • To use hash partitioning, ensure that each table has the same number of partitions. • To use range columns partitioning, reference the same number of columns in each table, and ensure that each table has the same number of partitions and uses the same partitioning column values. • To use range partitioning, ensure that each table has the same number of partitions and uses the same partitioning column values. • For sub-partitioning, follow the preceding rules based on the partitioning type.
opt_tg_partition_option	The partitioning rule of the table group. Use the partitioning rule that you specify when you create a table. A table group has no columns. Therefore, you do not need to specify column names for KEY, RANGE COLUMNS, and LIST COLUMNS. You only need to specify COLUMN_NUM.

Examples

- Create table group myTableGroup1.

```
OceanBase(admin@test)> CREATE TABLEGROUP myTableGroup1;
Query OK, 0 rows affected (0.07 sec)

OceanBase(admin@test)> create table myt1 (c1 int, c2 int ) tablegroup = myTableGroup1;
Query OK, 0 rows affected (0.28 sec)

OceanBase(admin@test)> create table myt2 (c1 int, c2 int ) tablegroup = myTableGroup1;
Query OK, 0 rows affected (0.26 sec)
```

- Create table group tgh and table ttgh. tgh and ttgh use hash partitioning and have the same number of partitions.

```
OceanBase(admin@test)> create tablegroup tgh locality='F,R{ALL_SERVER}@z1' partition by hash partitio
ns 10;
Query OK, 0 rows affected (0.09 sec)

OceanBase(admin@test)> create table ttgh(c1 int, c2 int) locality='F,R{ALL_SERVER}@z1' partition by h
ash(c1) partitions 10;
Query OK, 0 rows affected (0.55 sec)

OceanBase(admin@test)> create table ttgh2(c1 int, c2 int) locality='F,R{ALL_SERVER}@z1' partition by
hash(c2) partitions 10;
Query OK, 0 rows affected (0.39 sec)
```

17.1.4.5.19. CREATE TENANT

Description

You can execute the CREATE TENANT statement to create a tenant.

Syntax

```
CREATE TENANT [IF NOT EXISTS] tenantname
     [tenant_characteristic_list] [opt_set_sys_var]

tenant_characteristic_list:
tenant_characteristic [, tenant_characteristic...]

tenant_characteristic:
COMMENT 'string'
|{CHARACTER SET | CHARSET} [=] charsetname
|COLLATE [=]  collationname
|REPLICA_NUM [=] num
|ZONE_LIST [=] (zone [, zone…])
|PRIMARY_ZONE [=] zone
|DEFAULT TABLEGROUP [=] {NULL | tablegroup}
|RESOURCE_POOL_LIST [=](poolname [, poolname…])
|LOGONLY_REPLICA_NUM [=] num
|LOCALITY [=] 'locality description'

opt_set_sys_var:
{ SET | SET VARIABLES | VARIABLES } system_var_name = expr [,system_var_name = expr] ...
```

Parameters

Parameter	Description
tenant_name	The name of the tenant. The name must be a maximum of 64 bytes in length and can contain only letters, digits, and underscores (_). The name must start with a letter or an underscore (_) and cannot be a keyword that is reserved for ApsaraDB for OceanBase.
IF NOT EXISTS	If the specified tenant name is used and the IF NOT EXISTS option is added, the system returns an error.
RESOURCE_POOL_LIST	The resource pools. This parameter is required when you create a tenant. You can specify only one resource pool.
DEFAULT TABLEGROUP	The default table group of the tenant. A NULL value specifies no default table group for the database. If you do not specify this option, the default value is NULL.
COMMENT	The comments.
CHARACTER SET | CHARSET	The character set for the tenant.
COLLATE	The collation.
REPLICA_NUM	The number of replicas.
ZONE_LIST	The zones.
PRIMARY_ZONE	The primary zone.
LOGONLY_REPLICA_NUM	The number of log replicas.
LOCALITY	The distribution of replicas across zones. For example, the F@z1,F@z2,F@z3,R@z4 value specifies that replicas in the z1, z2, and z3 zones are FULL replicas, and replicas in the z4 zone are READONLY replicas.
system_var_name	The system variable of the tenant. The system variable ob_compatibility_mode specifies the compatibility mode of the tenant. You can set the tenant to be compatible with MySQL or Oracle. You can set the value only when you create a tenant. By default, the tenant is compatible with MySQL if you leave ob_compatibility_mode empty.

Examples

- Create a tenant.

```
CREATE TENANT IF NOT EXISTS t1 charset='utf8mb4', replica_num=1, zone_list=('zone1'), primary_zone='zone1', resource_pool_list=('pool1');
```

- Create a tenant that is compatible with Oracle.

```
CREATE TENANT IF NOT EXISTS t1 zone_list=('zone1'), primary_zone='zone1', resource_pool_list=('pool1') SET ob_compatibility_mode='oracle';
```

Notes

Before you execute the CREATE TENANT statement to create a tenant, connect your root user to the root tenant root@ROOT.

17.1.4.5.20. CREATE USER

Description

You can execute the CREATE USER statement to create a user in ApsaraDB for OceanBase. After a user is created, you can use the user to connect to ApsaraDB for OceanBase.

> **Note**
>
> To execute the CREATE USER statement, ensure you have the global CREATE USER permission.

Syntax

```
create_user_stmt:
    CREATE USER [IF NOT EXISTS] user_name [IDENTIFIED BY 'password'];

alter_user_stmt:
      ALTER USER user_name ACCOUNT {LOCK | UNLOCK};
    | ALTER USER user_name IDENTIFIED BY 'password';
    | SET PASSWORD [FOR user_name] = PASSWORD('password');
    | RENAME USER rename_user_action_list;

drop_user_stmt:
    DROP USER user_name_list;

rename_user_action_list:
    rename_user_action [, rename_user_action ...]

rename_user_action:
    user_name TO user_name

user_name_list:
    user_name [, user_name ...]

password:
    STR_VALUE

CREATE USER [IF NOT EXISTS] user_specification_list;

user_specification_list:
user_specification [, user_specification ...]

user_specification:
user IDENTIFIED BY 'authstring'
| user IDENTIFIED BY PASSWORD 'hashstring'
```

Parameters

Parameter	Description
user_name	The username. After a user is created, a new row is added for the user to the mysql.user table. If the username is used, the system returns an error.
IDENTIFIED BY	Set a password for the user.
user_name [, user_name ...]	To create multiple users at a time, separate them with commas (,).
user IDENTIFIED BY 'authstring'	The plaintext password. After the password is saved to the mysql.user table, the password is stored in ciphertext on the server.
user IDENTIFIED BY PASSWORD 'hashstring'	The ciphertext password.

Examples

1. Create users sqluser01 and sqluser02, and set their password to 123456.

```
CREATE USER 'sqluser01' IDENTIFIED BY '123456', 'sqluser02' IDENTIFIED BY '123456';
```

2. View the users you create.

```
SELECT user FROM mysql.user;
```

Output:

```
mysql> CREATE USER 'sqluser01' IDENTIFIED BY '123456', 'sqluser02' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.12 sec)
mysql> select user from mysql.user;
+-----------+
| user      |
+-----------+
| root      |
| admin     |
| sqluser01 |
| sqluser02 |
+-----------+
4 rows in set (0.00 sec)
```

17.1.4.5.21. CREATE VIEW

Description

You can execute the CREATE VIEW statement to create a view. If you add the OR REPLACE clause to the statement, you can execute the statement to replace an existing view.

Views are not stored as physical tables in databases. Views are generated each time you send a request to access the views. Views are created based on the outputs of the SELECT statements that are specified in the CREATE VIEW statements.

ApsaraDB for OceanBase V2.2.50 supports updatable views.

Syntax

```
create_view_stmt:
  CREATE [OR REPLACE] VIEW view_name [(column_name_list)] AS select_stmt;

column_name_list:
    column_name [, column_name ...]
```

Parameters

Parameter	Description
OR REPLACE	Create the view again if it exists. You can use this clause to change the definition of an existing view.
view_name	The name of the view.

Parameter	Description
select_stmt	The SELECT statement. This statement defines the view by selecting data from base tables or other views.
column_name_list	In views, column names must be unique. This rule for views is the same as that for base tables. By default, the column names that are returned by the SELECT statement are used as the column names for the view. To specify the column names of the view, use the optional column_name_list clause. You can use this clause to specify the column names. Separate column names with commas (,). The number of the column names in the column_name_list clause must be equal to the number of columns that are returned by the SELECT statement. The columns that are returned by the SELECT statement can be the specified table columns. The columns that are returned by the SELECT statement can also store the computing results of the expressions that use functions, constant values, or operators.

Examples

Create view v by using columns c1 and c2 in table t.

```
create or replace view v(vc1, vc2) as select c1, c2 from t;
```

17.1.4.5.22. DELETE

Description

You can execute the DELETE statement to delete rows that can satisfy specified conditions from one or more tables.

Syntax

```
Single-Table-Delete Syntax:
    DELETE [hint_options] FROM tbl_name
    [PARTITION (partition_name,...)]
    [WHERE where_condition]
    [ORDER BY order_expression_list]
    [LIMIT row_count]

Multiple-Table-Delete Syntax:
    DELETE [hint_options] tbl_name[.*] [, tbl_name[. *]] ...
    FROM table_references
    [WHERE where_condition]
Or:
    DELETE [hint_options] FROM tbl_name[.*] [, tbl_name[. *]] ...
    USING table_references
    [WHERE where_condition]

where_condition:
    expression

order_expression_list:
    order_expression [, order_expression ...]

order_expression:
    expression [ASC | DESC]

limit_row_count:
    INT_VALUE

table_references:
    {tbl_name | joined_table | table_subquery | select_with_parens} [, ...]
```

Parameters

Parameter	Description
hint_options	The hint.
tbl_name	The name of the table that you want to delete.
partition_name	The name of the table partition from which you want to delete data.
where_condition	The filter conditions. The system deletes the table rows that can satisfy the specified conditions.
order_expression_list	Sort the column values of the table from which you want to delete data. The values are sorted based on the sort keys.

Parameter	Description
row_count	The number of table rows that you want to delete. The value must be an integer.
table_references	The tables from which you want to delete data.

Examples

The following examples are based on tables t1 and t2.

```
OceanBase(admin@test)>create table t1(c1 int primary key, c2 int);
Query OK, 0 rows affected (0.16 sec)
OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |    1 |
|  2 |    2 |
|  3 |    3 |
|  4 |    4 |
+----+------+
4 rows in set (0.06 sec)

OceanBase(admin@test)>create table t2(c1 int primary key, c2 int) partition by key(c1) partitions 4;
Query OK, 0 rows affected (0.19 sec)
OceanBase(admin@test)>select * from t2;
+----+------+
| c1 | c2   |
+----+------+
|  5 |    5 |
|  1 |    1 |
|  2 |    2 |
|  3 |    3 |
+----+------+
4 rows in set (0.02 sec)
```

- Delete a row from a table: Delete a row where the value for the c1 column is equal to 2. The c1 column is the primary key of the t1 table.

```
OceanBase(admin@test)>DELETE FROM t1 WHERE c1 = 2;
Query OK, 1 row affected (0.02 sec)

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |    1 |
|  3 |    3 |
|  4 |    4 |
+----+------+
3 rows in set (0.01 sec)
```

- Delete a row from a table: Sort the rows in the t2 table by the c2 column, and delete the first row.

```
OceanBase(admin@test)>DELETE FROM t1 ORDER BY c2 LIMIT 1;
Query OK, 1 row affected (0.01 sec)

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  2 |    2 |
|  3 |    3 |
|  4 |    4 |
+----+------+
3 rows in set (0.00 sec)
```

- Delete a row from a table: Delete the p2 partition data from the t2 table.

```
OceanBase(admin@test)>DELETE FROM t2 PARTITION(p2);
Query OK, 3 rows affected (0.02 sec)

OceanBase(admin@test)>select * from t2;
+----+------+
| c1 | c2   |
+----+------+
|  5 |    5 |
+----+------+
1 row in set (0.02 sec)
```

- Delete rows from multiple tables: In the t1 and t2 tables, find the rows where the value for the c1 column in t1 is equal to that in t2, and delete the rows from t1 and t2.

```
OceanBase(admin@test)>DELETE t1, t2 FROM t1, t2 WHERE t1.c1 = t2.c1;
Query OK, 3 rows affected (0.02 sec)

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  4 |    4 |
+----+------+
1 row in set (0.01 sec)

OceanBase(admin@test)>select * from t2;
+----+------+
| c1 | c2   |
+----+------+
|  5 |    5 |
+----+------+
1 row in set (0.01 sec)
```

- Delete rows from multiple tables: In the t1 and t2 tables, find the rows where the value for the c1 column in t1 is equal to that in t2, and delete the rows from t1 and t2.

```
OceanBase(admin@test)>DELETE FROM t1, t2 USING t1, t2 WHERE t1.c1 = t2.c1;
Query OK, 4 rows affected (0.02 sec)

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  4 |    4 |
+----+------+
1 row in set (0.01 sec)

OceanBase(admin@test)>select * from t2;
Empty set (0.01 sec)
```

- Delete rows from multiple tables: In the t1 table and the p2 partition of the t2 table, find the rows where the value for the c1 column in t1 is equal to that in t2, and delete the rows from t1 and t2.

```
OceanBase(admin@test)>DELETE t2 FROM t1,t2 PARTITION(p2) WHERE t1.c1 = t2.c1;
Query OK, 3 rows affected (0.02 sec)

OceanBase(admin@test)>select * from t2;
+----+------+
| c1 | c2   |
+----+------+
|  5 |    5 |
+----+------+
1 row in set (0.01 sec)
```

- Delete a row from updatable view v.

```
OceanBase(admin@test)>create view v as select * from t1;
Query OK, 0 rows affected (0.07 sec)

OceanBase(admin@test)>delete from v where v.c1 = 1;
Query OK, 1 row affected (0.02 sec)

OceanBase(admin@test)>select * from v;
+----+------+
| c1 | c2   |
+----+------+
|  2 |    2 |
|  3 |    3 |
|  4 |    4 |
+----+------+
3 rows in set (0.01 sec)
```

Notes

You cannot use subqueries to delete rows from one or more tables. For example, you cannot execute the following statement:

```
delete from (select * from t1);
```

17.1.4.5.23. DROP DATABASE

Description

You can execute the DROP DATABASE statement to delete a database.

Syntax

```
drop_database_stmt:
    DROP DATABASE [IF EXISTS] database_name;
```

Parameters

Parameter	Description
IF EXISTS	Prevent errors that occur when you attempt to delete a database that does not exist.
database_name	The name of the database that you want to delete.

Examples

- Delete database test2.

```
OceanBase(admin@test)>drop database test2;
Query OK, 0 rows affected (0.03 sec)
```

- Delete database notest that does not exist.

```
OceanBase(admin@test)>drop database notest;
ERROR 1008 (HY000): Can't drop database 'notest'; database doesn't exist

OceanBase(admin@test)>drop database if exists notest;
Query OK, 0 rows affected, 1 warning (0.00 sec)

OceanBase(admin@test)>show warnings;
+-------+------+-----------------------------------------------------+
| Level | Code | Message                                             |
+-------+------+-----------------------------------------------------+
| Note  | 1008 | Can't drop database 'notest'; database doesn't exist |
+-------+------+-----------------------------------------------------+
1 row in set (0.00 sec)
```

17.1.4.5.24. DROP INDEX

Description

You can execute the DROP INDEX statement to delete an index. A large number of indexes require high maintenance overheads. We recommend that you delete the indexes that are not required.

After you execute the DROP INDEX statement to delete an index, the index cannot be immediately deleted. Wait for a period before the index is deleted.

Syntax

```
DROP INDEX indexname
     ON tblname;
```

Parameters

Parameter	Description
indexname	The name of the index.
tblname	The name of the table.

Examples

- Delete index test_index.

```
DROP INDEX test_index ON test;
```

17.1.4.5.25. DROP OUTLINE

Description

You can execute the DROP OUTLINE statement to delete an outline from an ApsaraDB for OceanBase database.

Syntax

```
DROP OUTLINE outline_name;
```

Parameters

Parameter	Description
outline_name	The name of the outline that you want to delete.

Examples

- Delete outline ol_1.

```
DROP OUTLINE ol_1;
```

17.1.4.5.26. DROP RESOURCE POOL

Description

You can execute the DROP RESOURCE POOL statement to delete a resource pool.

Syntax

```
DROP RESOURCE POOL poolname;
```

Parameters

Parameter	Description
poolname	The name of the resource pool that you want to delete.

Examples

- Delete resource pool pool1.

```
OceanBase(admin@test)> DROP RESOURCE POOL pool1;
Query OK, 0 rows affected (0.00 sec)
```

17.1.4.5.27. DROP RESOURCE UNIT

Description

You can execute the DROP RESOURCE UNIT statement to delete a resource unit.

Syntax

```
DROP RESOURCE UNIT unitname
```

Parameters

Parameter	Description
unitname	The name of the resource unit that you want to delete.

Examples

- Delete resource unit unit1.

```
OceanBase(admin@test)> DROP RESOURCE UNIT unit1;
Query OK, 0 rows affected (0.00 sec)
```

17.1.4.5.28. DROP TABLE

Description

You can execute the DROP TABLE statement to delete tables from an ApsaraDB for OceanBase database.

Syntax

```
DROP [TEMPORARY] {TABLE | TABLES} [IF EXISTS]
    table_name [,table_name]...
    [RESTRICT | CASCADE]
```

Parameters

Parameter	Description
table_name	The name of the table that you want to delete. To delete multiple tables at a time, separate the table names with commas (,).
IF EXISTS	If you add IF EXISTS to the statement and the table that you want to delete does not exist, the system does not return an error. If you do not add IF EXISTS to the statement and the table that you want to delete does not exist, the system returns an error.
TEMPORARY	Delete a temporary table.
RESTRICT | CASCADE	This parameter is required if you migrate data from other databases to ApsaraDB for OceanBase.

Examples

- Delete table test.

```
DROP TABLE IF EXISTS test;
```

17.1.4.5.29. DROP TABLEGROUP

Description

You can execute the DROP TABLEGROUP statement to delete a table group.

Syntax

```
DROP TABLEGROUP [IF EXISTS] tablegroupname
```

Parameters

Parameter	Description
tablegroupname	The name of the table group. Assume that the table group that you want to delete does not exist. If you do not use the IF EXISTS clause, the system returns an error.

Examples

Delete table group myTableGroup1.

```
OceanBase(admin@test)> DROP TABLEGROUP myTableGroup1;
```

17.1.4.5.30. DROP TENANT

Description

You can execute the DROP TENANT statement to delete a tenant from the ApsaraDB for OceanBase system.

Syntax

```
drop_tenant_stmt:
    DROP TENANT [IF EXISTS] tenant_name;
```

Parameters

Parameter	Description
tenant_name	The name of the tenant that you want to delete. Only a locked tenant can be deleted. If you execute this statement to delete an unlocked tenant, the system returns an error.

Examples

- Delete tenant TENANT1.

```
DROP TENANT TENANT1;
```

Notes

Before you execute the DROP TENANT statement to delete a tenant, connect your root user to the root tenant root@ROOT.

17.1.4.5.31. DROP SYNONYM

Description

You can execute the DROP SYNONYM statement to delete a synonym.

Syntax

```
DROP [PUBLIC] SYNONYM [ DATABASE. ]synonym;
```

Parameters

- PUBLIC

  Delete a public synonym. If you do not use PUBLIC, a private synonym is deleted.

- [DATABASE.]synonym

  The database to which the current synonym belongs. If you add PUBLIC to the statement, you do not need to specify the database for the synonym. synonym specifies the name of the synonym.

Examples

- Delete a synonym.

```
OceanBase(admin@test)>drop synonym test.s1;
Query OK, 0 rows affected (0.03 sec)
```

- Delete a public synonym.

```
OceanBase(admin@test)>drop public synonym syn_pub;
Query OK, 0 rows affected (0.02 sec)
```

Notes

Notice

- To delete a private synonym, ensure that the synonym belongs to the specified database and you have the DROP ANY SYNONYM permission.
- To delete a public synonym, ensure that you have the DROP PUBLIC SYNONYM permission.
- To delete a public synonym, make sure that you add PUBLIC to the statement and do not specify the database.

17.1.4.5.32. DROP USER

Description

You can execute the DROP USER statement to delete one or more users from ApsaraDB for OceanBase.

Note

- To execute the DROP USER statement, ensure that you have the global permission to execute the CREATE USER statement.
- You cannot execute a DELETE statement to manage permissions on the mysql.user table.
- After a user is deleted, all permissions of the user are removed.

Syntax

```
DROP USER username [, username...] ;
```

Parameters

Parameter	Description
username	The username. To delete multiple users at a time, separate the usernames with commas (,).

Examples

Run the following command to delete user sqluser02:

```
oceanBase(admin@TEST)>drop user sqluser02;
Query OK, 0 rows affected (0.02 sec)
```

17.1.4.5.33. DROP VIEW

Description

You can execute the DROP VIEW statement to delete one or more views.

> **Note**
>
> Before you execute this statement, ensure you have the global DROP permission.

Syntax

```
drop_view_stmt:
    DROP VIEW [IF EXISTS] view_name_list [CASCADE | RESTRICT];

view_name_list:
    view_name [, view_name_list]
```

Parameters

Parameter	Description
IF EXISTS	If you add IF EXISTS to the statement and the specified views do not exist, the system does not return an error.
view_name_list	If some views you specify in view_name_list do not exist, the system may return an error. However, the system still deletes the views that exist.
CASCADE | RESTRICT	CASCADE and RESTRICT are parsed and ignored.

Examples

Execute the following statements to delete views v1 and v2: Assume that v1 or v2 does not exist. If you execute the first statement, the system returns an error.

```
drop view v1, v2;

drop view if exists v1, v2;
```

17.1.4.5.34. EXPLAIN

Description

You can execute the EXPLAIN statement to query the execution plans for SQL statements. EXPLAIN supports the SELECT, DELETE, INSERT, REPLACE, and UPDATE statements.

Syntax

```
Query a table or a column:
{EXPLAIN | DESCRIBE | DESC} tbl_name [col_name | wild]

Query the execution plan for an SQL statement:
{EXPLAIN | DESCRIBE | DESC}
[BASIC | OUTLINE | EXTENDED | EXTENDED_NOADDR | PARTITIONS | FORMAT = {TRADITIONAL| JSON}]
{SELECT statement | DELETE statement | INSERT statement | REPLACE statement| UPDATE statement}
```

Parameters

Parameter	Description
tbl_name	The name of the table.
col_name	The name of the table column.
BASIC	Query the basic information about the execution plan, such as the operator IDs, the operator names, and the names of the tables that are referenced.
OUTLINE	Query the information about the execution plan, including the outline.
EXTENDED	Query the additional information generated by the EXPLAIN statement. The information includes the input and output columns of each operator, the partitions of the referenced tables, and the specified filter conditions. If the current operator uses an index, the system returns the index columns and query range.
EXTENDED_NOADDR	Display the additional information in a clear way.
PARTITIONS	Query the partition information.
FORMAT = {TRADITIONAL| JSON}	The output format of the EXPLAIN statement. Valid values: • TRADITIONAL: displays the output in a tabular format. • JSON: displays the output in a JSON string that contains key-value pairs. The output information includes the additional information and partition information.

Examples

- **Leave the parameters that specify the output type empty.**

```
OceanBase(admin@test)>explain select * from t1,t2 where t1.c2=t2.c2 and t2.c1 > 4\G
*************************** 1. row ***************************
Query Plan: ========================================
|ID|OPERATOR   |NAME|EST. ROWS|COST   |
----------------------------------------
|0 |HASH JOIN  |    |9801000  |5933109|
|1 | TABLE SCAN|t2  |10000    |6219   |
|2 | TABLE SCAN|t1  |100000   |68478  |
========================================

Outputs & filters:
-------------------------------------
  0 - output([t1.c1], [t1.c2], [t2.c1], [t2.c2]), filter(nil),
      equal_conds([t1.c2 = t2.c2]), other_conds(nil)
  1 - output([t2.c2], [t2.c1]), filter(nil),
      access([t2.c2], [t2.c1]), partitions(p0)
  2 - output([t1.c2], [t1.c1]), filter(nil),
      access([t1.c2], [t1.c1]), partitions(p0)
```

- **EXTENDED**

```
OceanBase(admin@test)>explain extended_noaddr select * from t1,t2 where t1.c2=t2.c2 and t2.c1 > 4\G
*************************** 1. row ***************************
Query Plan: ========================================
|ID|OPERATOR   |NAME|EST. ROWS|COST   |
----------------------------------------
|0 |HASH JOIN  |    |9801000  |5933109|
|1 | TABLE SCAN|t2  |10000    |6219   |
|2 | TABLE SCAN|t1  |100000   |68478  |
========================================

Outputs & filters:
-------------------------------------
  0 - output([t1.c1], [t1.c2], [t2.c1], [t2.c2]), filter(nil),
      equal_conds([t1.c2 = t2.c2]), other_conds(nil)
  1 - output([t2.c2], [t2.c1]), filter(nil),
      access([t2.c2], [t2.c1]), partitions(p0),
      is_index_back=false,
      range_key([t2.c1]), range(4 ; MAX),
      range_cond([t2.c1 > 4])
  2 - output([t1.c2], [t1.c1]), filter(nil),
      access([t1.c2], [t1.c1]), partitions(p0),
      is_index_back=false,
      range_key([t1.__pk_increment], [t1.__pk_cluster_id], [t1.__pk_partition_id]), range(MIN,MIN,MIN
; MAX,MAX,MAX)always true
```

- **TRADITIONAL format**

```
OceanBase(admin@test)>explain format=TRADITIONAL select * from t1,t2 where t1.c2=t2.c2 and t2.c1 > 4\
G
*************************** 1. row ***************************
Query Plan: ========================================
|ID|OPERATOR   |NAME|EST. ROWS|COST   |
----------------------------------------
|0 |HASH JOIN  |    |9801000  |5933109|
|1 | TABLE SCAN|t2  |10000    |6219   |
|2 | TABLE SCAN|t1  |100000   |68478  |
========================================

Outputs & filters:
-------------------------------------
  0 - output([t1.c1], [t1.c2], [t2.c1], [t2.c2]), filter(nil),
      equal_conds([t1.c2 = t2.c2]), other_conds(nil)
  1 - output([t2.c2], [t2.c1]), filter(nil),
      access([t2.c2], [t2.c1]), partitions(p0)
  2 - output([t1.c2], [t1.c1]), filter(nil),
      access([t1.c2], [t1.c1]), partitions(p0)
```

- **JSON format**

```
OceanBase(admin@test)>explain format=JSON select * from t1,t2 where t1.c2=t2.c2 and t2.c1 > 4\G
*************************** 1. row ***************************
Query Plan: {
  "ID":2,
  "OPERATOR":"JOIN",
  "NAME":"JOIN",
  "EST.ROWS":9800999,
  "COST":5933108,
  "output": [
    "t1.c1",
    "t1.c2",
    "t2.c1",
    "t2.c2"
  ],
  "TABLE SCAN": {
    "ID":0,
    "OPERATOR":"TABLE SCAN",
    "NAME":"TABLE SCAN",
    "EST.ROWS":10000,
    "COST":6218,
    "output": [
      "t2.c2",
      "t2.c1"
    ]
  },
  "TABLE SCAN": {
    "ID":1,
    "OPERATOR":"TABLE SCAN",
    "NAME":"TABLE SCAN",
    "EST.ROWS":100000,
    "COST":68477,
    "output": [
      "t1.c2",
      "t1.c1"
    ]
  }
}
```

Each row in the output of the EXPLAIN statement contains the following columns.

Column	Description
ID	The sequential number of the execution plan.
OPERATOR	The operator in the execution plan.
NAME	The table referenced by the operator.
EST.ROWS	The estimated number of rows in the output of the operator.
COST	The CPU time consumed by the operator.

17.1.4.5.35. FLASHBACK DATABASE

Description

You can execute the FLASHBACK DATABASE statement to restore a deleted database from the recycle bin.

Prerequisites

The recycle bin is enabled. You can execute the following statement to check whether the recycle bin is enabled:

```
show variables like 'recyclebin';
```

```
OceanBase(admin@test)> show variables like 'recyclebin';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| recyclebin    | ON    |
+---------------+-------+
1 row in set (0.00 sec)
```

If the recycle bin is disabled, you can execute the following statement to enable it:

```
set recyclebin = on;
```

Databases in the recycle bin are not deleted from the system and still consume resources. To delete them from the system, execute the following statement:

```
purge recyclebin;
```

Syntax

```
 FLASHBACK DATABASE object_name TO BEFORE DROP [RENAME TO db_name];
```

Parameters

Parameter	Description
object_name	The name of the object that you want to restore. You cannot specify a database name. If you restore a database, the tables and indexes in the database are also restored.
RENAME to	Rename the database that you want to restore.

Examples

- Restore a deleted database from the recycle bin.

```
OceanBase(admin@test)> create database da;
Query OK, 1 row affected (0.03 sec)

OceanBase(admin@test)> drop database da;
Query OK, 0 rows affected (0.04 sec)

OceanBase(admin@test)> show recyclebin;
+-------------------------------------------------+---------------+----------+---------------------
------+
| OBJECT_NAME                                     | ORIGINAL_NAME | TYPE     | CREATETIME
|
+-------------------------------------------------+---------------+----------+---------------------
------+
| __recycle_$_1_1099511628829_18446744073709551615 | da            | DATABASE | 2017-10-20 17:36:15.8
38771 |
+-------------------------------------------------+---------------+----------+---------------------
------+
1 row in set (0.02 sec)

OceanBase(admin@test)> flashback database __recycle_$_1_1099511628829_18446744073709551615 to before
drop;
Query OK, 0 rows affected (0.03 sec)
```

17.1.4.5.36. FLASHBACK TABLE

Description

You can execute the FLASHBACK TABLE statement to restore a deleted table from the recycle bin.

Prerequisites

The recycle bin is enabled. You can execute the following statement to check whether the recycle bin is enabled:

```
show variables like 'recyclebin';
```

```
OceanBase(admin@test)> show variables like 'recyclebin';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| recyclebin    | ON    |
+---------------+-------+
1 row in set (0.00 sec)
```

If the recycle bin is disabled, you can execute the following statement to enable it:

```
set recyclebin = on;
```

Tables in the recycle bin are not deleted from the database and still consume resources. To delete them from the database, execute the following statement:

```
purge recyclebin;
```

Syntax

```
 FLASHBACK TABLE object_name TO BEFORE DROP [RENAME to db_name.table_name];
```

Parameters

Parameter	Description
object_name	The name of the object or table that you want to restore. Before you execute FLASHBACK TABLE to restore a table, log on to the database where the table is stored. When you restore a table, the index of the table is also restored.
RENAME to	Rename the table and database to which the table belongs.

Examples

- Restore deleted table t from the recycle bin.

```
OceanBase(admin@test)> create table t(id int primary key, k int);
Query OK, 0 rows affected (0.04 sec)

OceanBase(admin@test)> insert into t values(1,1);
Query OK, 1 row affected (0.00 sec)

OceanBase(admin@test)> select * from t;
+----+------+
| id | k    |
+----+------+
|  1 |    1 |
+----+------+
1 row in set (0.00 sec)

OceanBase(admin@test)> drop table t;
Query OK, 0 rows affected (0.01 sec)

OceanBase(admin@test)> select * from t;
ERROR 1146 (42S02): Table 'test.t' does not exist
OceanBase(admin@test)> show recyclebin;
+--------------------------------+---------------+-------+----------------------------+
| OBJECT_NAME                    | ORIGINAL_NAME | TYPE  | CREATETIME                 |
+--------------------------------+---------------+-------+----------------------------+
| __recycle_$_1_1597028971700936 | t             | TABLE | 2020-08-10 11:09:31.701033 |
+--------------------------------+---------------+-------+----------------------------+
1 row in set (0.00 sec)

OceanBase(admin@test)> flashback table t to before drop;
Query OK, 0 rows affected (0.01 sec)

OceanBase(admin@test)> select * from t;
+----+------+
| id | k    |
+----+------+
|  1 |    1 |
+----+------+
1 row in set (0.00 sec)
```

17.1.4.5.37. GRANT

Description

You can execute the GRANT statement to grant permissions to a user as a system administrator.

> **Note**
>
> - Before you grant permissions to a user, ensure that you have the permissions. For example, to grant User 2 the SELECT permission on Table 1 as User 1, ensure that User 1 has the permission to be granted and the GRANT OPTION permission. Otherwise, you cannot grant the permission to User 2.
> - After you grant a permission to a user, the user must log on to the ApsaraDB for OceanBase system again to ensure that the permission takes effect.

Syntax

```
GRANT priv_type
       ON priv_level
       TO user_specification [, user_specification]...
     [WITH with_option ...]

privilege_type:
      ALTER
    | CREATE
    | CREATE USER
    | CREATE VIEW
    | DELETE
    | DROP
    | GRANT OPTION
    | INDEX
    | INSERT
    | PROCESS
    | SELECT
    | SHOW DATABASES
    | SHOW VIEW
    | SUPER
    | UPDATE
    | USAGE
    | CREATE SYNONYM

priv_level:
     *
    | *.*
    | db_name.*
    | db_name.tbl_name
    | tbl_name
    | db_name.rountine_name

user_specification:
user [IDENTIFIED BY [PASSWORD] 'Password']

with_option:
 GRANT OPTION
```

Parameters

Parameter	Description
priv_type	The permission to be granted. For more information, see the Permissions table in the following description. To grant multiple permissions to a user at a time, separate the permissions with commas (,).
priv_level	The level of the permission to be granted. Permissions can be divided into the following levels: • Global level: The permissions apply to all the databases. Use GRANT ALL ON *.* to grant global permissions. • Database level: The permissions apply to all the objects in the specified database. Use GRANT ALL ON db_name.* to assign database-level permissions. • Table level: The permissions apply to all the columns in the specified table. Use GRANT ALL ON db_name.tbl_name to assign table-level permissions. You can use an asterisk (*) instead of specifying a table name to assign permissions on all tables in the specified database.
user_specification	Grant permissions to the specified user. If the specified user does not exist, the system can create a user. If sql_mode is set to no_auto_create_user and you do not specify a password by using IDENTIFIED BY, the system does not automatically create a user. To grant permissions to multiple users at a time, separate the usernames with commas (,).
user IDENTIFIED BY 'Password'	Specifies a plaintext password.
user IDENTIFIED BY PASSWORD 'Password'	Specifies a ciphertext password.
with_option	Specifies whether to allow authorized users to grant permissions to other users.

The following table lists the permissions that can be assigned.

Permissions

Permission	Description
ALL PRIVILEGES	All permissions except GRANT OPTION.
ALTER	The permission to execute the ALTER TABLE statement.

Permission	Description
CREATE	The permission to execute the CREATE TABLE statement.
CREATE USER	The permission to execute the CREATE USER, DROP USER, RENAME USER, and REVOKE ALL PRIVILEGES statements.
CREATE TABLEGROUP	The global permission to execute the CREATE TABLEGROUP statement.
DELETE	The permission to execute the DELETE statement.
DROP	The permission to execute the DROP statement.
GRANT OPTION	The permission to execute the GRANT OPTION statement.
INSERT	The permission to execute the INSERT statement.
SELECT	The permission to execute the SELECT statement.
UPDATE	The permission to execute the UPDATE statement.
SUPER	The permission to execute the SET GLOBAL statement to modify global system parameters.
SHOW DATABASES	The global permission to execute the SHOW DATABASES statement.
INDEX	The permission to execute the CREATE INDEX and DROP INDEX statements.
CREATE VIEW	The permission to create or delete a view.
SHOW VIEW	The permission to execute the SHOW CREATE VIEW statement.
CREATE SYNONYM	The permission to create a synonym.

Note

The permission to execute the CHANGE EFFECTIVE TENANT statement is not limited. Therefore, all users under the system tenant can grant this permission to other users.

Examples

- Run the following command to grant ALL PRIVILEGES to the obsqluser01 user:

```
OceanBase(admin@TEST)>GRANT ALL PRIVILEGES ON *. * TO obsqluser01 with grant option;
Query OK, 0 rows affected (0.03 sec)
```

17.1.4.5.38. INSERT

Description

You can execute the INSERT statement to add one or more records to a table.

Syntax

```
INSERT [IGNORE] [INTO]
       single_table_insert
       [ON DUPLICATE KEY UPDATE update_asgn_list]

single_table_insert:
    {dml_table_name values_clause
     | dml_table_name '(' ')' values_clause
     | dml_table_name '(' column_list ')' values_clause
     | dml_table_name SET update_asgn_list}

dml_table_name:
    tbl_name [PARTITION (partition_name,...)]

values_clause:
     {{VALUES | VALUE} ({expr | DEFAULT},...) [, ...]
      | select_stmt}

column_list
    column_name [, ...]

update_asgn_list:
     column_name = expr [, ...]
```

Parameters

If you execute the INSERT...ON DUPLICATE KEY UPDATE... statement, the number of affected rows is calculated based on the following rules:

- If you do not specify the CLIENT_FOUND_ROWS flag in client_capabilities, the number of affected rows is calculated based on the following rules:
  - If a new row is inserted, one row is affected.
  - If an inserted row conflicts with an existing row in the table and the data in the table remains the same after the update, no row is affected. If the data in the table before the update is different from that after the update, two rows are affected.
- If you specify the CLIENT_FOUND_ROWS flag, the number of affected rows is calculated based on the following rules:
  - If a new row is inserted, one row is affected.
  - If the data in the table remains the same after the update, one row is affected.
  - If the data in the table before the update is different from that after the update, two rows are affected.
- If you do not specify the CLIENT_FOUND_ROWS flag, the number of affected rows equals the number of

updated rows. If you specify the CLIENT_FOUND_ROWS flag, the number of affected rows equals the number of touched rows that conflict with existing rows. The data in the touched rows may not be modified.

Parameter	Description
IGNORE	Ignore the errors that occur when the INSERT statement is executed.
column_list	The columns to insert. Separate multiple columns with commas (,).
tbl_name	The name of the table to insert.
partition_name	The partition name of the table to insert.
update_asgn_list	Assign values to the column. For example, c1 = 2.
ON DUPLICATE KEY UPDATE	Specifies whether to update duplicate primary key or unique key values. Assume that a value you insert conflicts with an existing primary key or unique key value. In this case, if you have added ON DUPLICATE KEY UPDATE to the statement, the new value overwrites the existing value. If you do not add ON DUPLICATE KEY UPDATE, the system returns an error when duplicate primary key or unique key values exist.

Examples

The following examples are based on tables t1 and t2.

```
OceanBase(admin@test)>create table t1(c1 int primary key, c2 int) partition by key(c1) partitions 4;
Query OK, 0 rows affected (0.16 sec)

OceanBase(admin@test)>create table t2(c1 int primary key, c2 int);
Query OK, 0 rows affected (0.16 sec)
OceanBase(admin@test)>select * from t2;
+----+------+
| c1 | c2   |
+----+------+
|  1 |    1 |
|  2 |    2 |
|  3 |    3 |
|  4 |    4 |
+----+------+
4 rows in set (0.06 sec)
```

- Insert a row into table t1.

```
OceanBase(admin@test)>insert into t1 values(1,1);
Query OK, 1 row affected (0.01 sec)

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |    1 |
+----+------+
1 row in set (0.04 sec)
```

- Insert multiple rows into table t1.

```
OceanBase(admin@test)>insert t1 values(1,1),(2,default),(2+2,3*4);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |    1 |
|  2 | NULL |
|  4 |   12 |
+----+------+
3 rows in set (0.02 sec)
```

- Insert a row into partition p0 in table t1.

```
OceanBase(admin@test)>insert into t1 partition(p0) (c1) values(5);
Query OK, 1 row affected (0.02 sec)
OceanBase(admin@test)>select * from t1 partition(p0);
+----+------+
| c1 | c2   |
+----+------+
|  5 | NULL |
+----+------+
1 row in set (0.01 sec)
```

- Insert the query results from table t2 into table t1.

```
OceanBase(admin@test)>insert into t1 select * from t2;
Query OK, 4 rows affected (0.02 sec)
Records: 4  Duplicates: 0  Warnings: 0

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |    1 |
|  2 |    2 |
|  3 |    3 |
|  4 |    4 |
+----+------+
4 rows in set (0.01 sec)
```

- Insert data into table t1 and add ON DUPLICATE KEY UPDATE to the statement to update duplicate primary key

values.

```
OceanBase(admin@test)>insert into t1 values(1,1),(1,2) ON DUPLICATE KEY UPDATE c1=100;
Query OK, 3 rows affected (0.01 sec)
Records: 2  Duplicates: 1  Warnings: 0

OceanBase(admin@test)>select * from t1;
+-----+------+
| c1  | c2   |
+-----+------+
| 100 |    1 |
+-----+------+
1 row in set (0.02 sec)
```

- Insert data into updatable view v.

```
OceanBase(admin@test)>create view v as select * from t1;
Query OK, 0 rows affected (0.07 sec)
OceanBase(admin@test)>insert into v values(1,1);
Query OK, 1 row affected (0.01 sec)

OceanBase(admin@test)>select * from v;
+----+------+
| c1 | c2   |
+----+------+
|  1 |    1 |
+----+------+
1 row in set (0.02 sec)
```

Notes

You cannot use subqueries in the INSERT statement to insert rows. For example, you cannot execute the following statement:

```
insert into (select * from t1) values(1, 1);
```

17.1.4.5.39. KILL

Description

You can execute the KILL statement to close a session.

> **Note**
>
> If you have the PROCESS permission, you can view all sessions. If you have the SUPER permission, you can close all sessions and statements. If you do not have the PROCESS or SUPER permission, you can view or close only your own sessions and statements.

Syntax

```
KILL [CONNECTION | QUERY] 'sessionid'
```

Parameters

Parameter	Description
KILL CONNECTION	This statement performs the same operation as a KILL statement that does not contain a modifier. This statement closes the connection that is associated with the specified *sessionid*.
KILL QUERY	Stop the execution of statements that are being executed in the connection, but leaves the connection intact.

Examples

Stop the execution of the statement that is being executed in session 3221638213, and then close the session.

```
OceanBase(admin@test)>show processlist;
+------------+-------+--------------------+------+---------+------+--------+------------------+
| Id         | User  | Host               | db   | Command | Time | State  | Info             |
+------------+-------+--------------------+------+---------+------+--------+------------------+
| 3221638212 | admin | 1.11.111.127:11161 | test | Query   |    0 | ACTIVE | show processlist |
| 3221638213 | admin | 1.11.111.127:11161 | test | Query   |    0 | ACTIVE | select "abcedfg" |
+------------+-------+--------------------+------+---------+------+--------+------------------+
1 row in set (0.03 sec)

OceanBase(admin@test)>kill query 3221638213;
Query OK, 0 rows affected (0.01 sec)

OceanBase(admin@test)>kill 3221638212;
Query OK, 0 rows affected (0.01 sec)
```

17.1.4.5.40. PURGE DATABASE

Description

You can execute the PURGE DATABASE statement to delete a database from the recycle bin.

Syntax

```
PURGE DATABASE object_name;
```

Parameters

Parameter	Description
object_name	The name of the object in the recycle bin. After an object is moved to the recycle bin, the system renames the object.

Examples

- Delete database __recycle_$_1_1597384386029184 from the recycle bin.

```
OceanBase(admin@test)> create database db1;
Query OK, 1 row affected (0.03 sec)

OceanBase(admin@test)> drop database db1;
Query OK, 0 rows affected (0.04 sec)

OceanBase(admin@test)> show recyclebin;
+--------------------------------+---------------+----------+----------------------------+
| OBJECT_NAME                    | ORIGINAL_NAME | TYPE     | CREATETIME                 |
+--------------------------------+---------------+----------+----------------------------+
| __recycle_$_1_1597384386029184 | db1           | DATABASE | 2020-08-14 13:53:06.029367 |
+--------------------------------+---------------+----------+----------------------------+
1 row in set (0.01 sec)

OceanBase(admin@test)> purge database __recycle_$_1_1597384386029184;
Query OK, 0 rows affected (0.03 sec)

OceanBase(admin@test)> show recyclebin;
```

17.1.4.5.41. PURGE INDEX

Description

You can execute the PURGE INDEX statement to delete an indexed table from the recycle bin.

Syntax

```
PURGE INDEX object_name;
```

Parameters

Parameter	Description
object_name	The name of the object in the recycle bin. After an object is moved to the recycle bin, the system renames the object.

Examples

- Delete indexed table __recycle_$_1_1597387726700872 from the recycle bin.

```
OceanBase(admin@test)> create table t1(c1 int);
Query OK, 0 rows affected (0.09 sec)

OceanBase(admin@test)> create index idx on t1(c1);
Query OK, 0 rows affected (0.48 sec)

OceanBase(admin@test)> drop table t1;
Query OK, 0 rows affected (0.03 sec)

OceanBase(admin@test)> show recyclebin;
+--------------------------------+----------------------------+-------+----------------------------+
| OBJECT_NAME                    | ORIGINAL_NAME              | TYPE  | CREATETIME                 |
+--------------------------------+----------------------------+-------+----------------------------+
| __recycle_$_1_1597387726700872 | __idx_1101710651081557_idx | INDEX | 2020-08-14 14:48:46.699145 |
| __recycle_$_1_1597387726712976 | t1                         | TABLE | 2020-08-14 14:48:46.712643 |
+--------------------------------+----------------------------+-------+----------------------------+
5 rows in set (0.01 sec)

OceanBase(admin@test)> purge index __recycle_$_1_1597387726700872;
Query OK, 0 rows affected (0.04 sec)
```

17.1.4.5.42. PURGE RECYCLEBIN

Description

You can execute the PURGE RECYCLEBIN statement to empty the recycle bin as the root user.

> **Notice**
>
> The PURGE RECYCLEBIN statement clears all objects in the recycle bin. You can execute this statement only as the root user. Proceed with caution.

Syntax

```
PURGE RECYCLEBIN;
```

Parameters

None.

Examples

- Empty the recycle bin as the root user.

```
OceanBase(admin@test)> purge recyclebin;
Query OK, 0 rows affected (0.03 sec)
```

17.1.4.5.43. PURGE TABLE

Description

You can execute the PURGE TABLE statement to delete a table from the recycle bin.

Syntax

```
PURGE TABLE object_name;
```

Parameters

Parameter	Description
object_name	The name of the object in the recycle bin. After an object is moved to the recycle bin, the system renames the object.

Examples

- Delete table __recycle_$_1_1099511628776_1099511677778 from the recycle bin.

```
OceanBase(admin@test)> create table test(c1 int);
Query OK, 0 rows affected (0.16 sec)

OceanBase(admin@test)> drop table test;
Query OK, 0 rows affected (0.03 sec)

OceanBase(admin@test)> show recyclebin;
+-------------------------------------------+---------------+-------+----------------------------+
| OBJECT_NAME                               | ORIGINAL_NAME | TYPE  | CREATETIME                 |
+-------------------------------------------+---------------+-------+----------------------------+
| __recycle_$_1_1099511628776_1099511677778 | test          | TABLE | 2017-10-20 17:40:22.304025 |
+-------------------------------------------+---------------+-------+----------------------------+
1 row in set (0.02 sec)

OceanBase(admin@test)> purge table __recycle_$_1_1099511628776_1099511677778;
Query OK, 0 rows affected (0.04 sec)
```

17.1.4.5.44. RENAME TABLE

Description

You can execute the RENAME TABLE statement to rename one or more tables.

Syntax

```
RENAME TABLE tblname TO newtblname
     [, tblname2 TO newtblname …] ;
```

Parameters

Parameter	Description
tblname	The name of the original table.
newtblname	The name of the new table.

Parameter	Description
tblname TO newtblname [, tb1name2 TO newtblname ...] ;	To rename multiple tables at a time, separate the table names with commas (,).

Notes

- After you execute the RENAME TABLE statement, the specified tables are automatically renamed. When the tables are being renamed, other threads cannot read data from the tables.
- If you execute this statement to rename multiple tables, the tables are renamed in the left-to-right order.
- To execute the RENAME TABLE statement, ensure that no tables are locked or involved in active transactions. To execute the RENAME TABLE statement, ensure that you have the ALTER and DROP permissions on the original table and the CREATE and INSERT permissions on the new table.
- You can execute the RENAME TABLE statement to rename views.

Examples

1. Create tables t1 and t2.

```
create table t1(c1 int);

create table t2(c1 int);
```

2. Rename table t1 to t11.

```
rename table t1 to t11;
```

3. Rename table t11 to t111 and table t2 to t22.

```
rename table t11 to t111, t2 to t22;
```

4. Rename table t111 to t1111, and then rename table t1111 to t1.

```
rename table t111 to t1111, t1111 to t1;
```

17.1.4.5.45. RENAME USER

Description

You can execute the RENAME USER statement to change the username that is used to log on to the ApsaraDB for OceanBase system.

> **Note**
>
> To execute this statement, ensure you have the global CREATE USER permission.

Syntax

```
RENAME USER
'oldusername' TO 'newusername'
[,'oldusername' TO 'newusername'...] ;
```

Parameters

Parameter	Description
oldusername	The old username.
newusername	The new username. The name must be a maximum of 16 bytes in length.
'oldusername' TO 'newusername'	Changes the username. To change multiple usernames at a time, separate them with commas (,). After you change the username, the permissions of the user remain unchanged.

Examples

1. Before you change a username, execute the following statement to view users:

```
SELECT user FROM mysql.user;
```

Output:

```
mysql> SELECT user FROM mysql.user;
+-----------+
| user      |
+-----------+
| root      |
| admin     |
| sqluser01 |
| sqluser02 |
+-----------+
4 rows in set (0.00 sec)
```

2. Execute the following statement to change username sqluser01 to obsqluser01.

```
RENAME USER 'sqluser01' TO 'obsqluser01';
```

3. After you change the username, execute the following statement to view users:

```
SELECT user FROM mysql.user;
```

Output: Username sqluser01 is changed to obsqluser01.

```
mysql> RENAME USER 'sqluser01' TO 'obsqluser01';
Query OK, 0 rows affected (0.04 sec)

mysql> SELECT user FROM mysql.user;
+-------------+
| user        |
+-------------+
| root        |
| admin       |
| obsqluser01 |
| sqluser02   |
+-------------+
4 rows in set (0.00 sec)
```

17.1.4.5.46. REPLACE

Description

You can execute the REPLACE statement to replace one or more records in a table. If an old record has the same value as a new record in a primary or unique key column, the old record is deleted before the new record is inserted.

Syntax

```
replace_stmt:
    REPLACE [INTO] table_factor [PARTITION (partition_name_list)] [(column_name_list)]
    {VALUES | VALUE} column_value_lists;

partition_name_list:
        partition_name [, partition_name ...]

column_name_list:
    column_name [, column_name ...]

column_value_lists:
        (column_value_list) [, (column_value_list) ...]

column_value_list:
    column_value [, column_value ...]

column_value:
    {expression | DEFAULT}
```

Parameters

Parameter	Description
table_factor	The name of the table in which you want to replace records.
column_name_list	The columns in which you want to replace data.

Parameter	Description
partition_name_list	The table partitions in which you want to replace records.

Examples

The following statement defines a table named test. All the examples in this topic are based on the test table.

```
OceanBase(admin@test)>create table test (c1 int primary key, c2 varchar(40));
Query OK, 0 rows affected (0.23 sec)
```

1. Replace the value in row 1 in the test table with 'hello alibaba' and replace the value in row 2 with 'hello ob'.

```
OceanBase(admin@test)>REPLACE INTO test VALUES (1, 'hello alibaba'),(2, 'hello ob');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

2. Query row 1 and row 2 in the test table.

```
OceanBase(admin@test)>SELECT * FROM test;
+----+---------------+
| c1 | c2            |
+----+---------------+
|  1 | hello alibaba |
|  2 | hello ob      |
+----+---------------+
2 rows in set (0.00 sec)
```

3. Replace the value in row 3 in the test table with 'hello alibaba' and replace the value in row 2 with 'hello oceanbase'.

```
OceanBase(admin@test)>REPLACE INTO test VALUES (3, 'hello alibaba'),(2, 'hello oceanbase');
Query OK, 3 rows affected (0.00 sec)
Records: 2  Duplicates: 1  Warnings: 0
```

4. Query row 1, row 2, and row 3 in the test table.

```
OceanBase(admin@test)>SELECT * FROM test;
+----+-----------------+
| c1 | c2              |
+----+-----------------+
|  1 | hello alibaba   |
|  2 | hello oceanbase |
|  3 | hello alibaba   |
+----+-----------------+
3 rows in set (0.00 sec)
```

17.1.4.5.47. REVOKE

Description

You can execute the REVOKE statement to revoke permissions from users as a system administrator.

Notes:

- To revoke permissions from a user, ensure that you have the permissions to be revoked. For example, to revoke the SELECT permission on Table 1 from User 2 as User 1, ensure that User 1 has this permission and the GRANT OPTION permission.
- To revoke the ALL PRIVILEGES or GRANT OPTION permission, ensure that you have the global permission to execute the GRANT OPTION statement or the UPDATE and DELETE permissions on the permission table.
- Revocations do not have cascading effects. Assume that User 1 has granted permissions to User 2. If you revoke permissions from User 1, the permissions are not revoked from User 2.

Syntax

```
REVOKE priv_type
         ON database.tblname
         FROM 'username';

privilege_type:
      ALTER
    | CREATE
    | CREATE USER
    | CREATE VIEW
    | DELETE
    | DROP
    | GRANT OPTION
    | INDEX
    | INSERT
    | PROCESS
    | SELECT
    | SHOW DATABASES
    | SHOW VIEW
    | SUPER
    | UPDATE
    | USAGE
```

Parameters

Parameter	Description
priv_type	The permission that you want to revoke. For more information, see the Permissions table in the following description. To revoke multiple permissions from a user, separate the permissions with commas (,).
database.tblname	The table in the database. To revoke permissions on all tables in the database, use an asterisk (*) to replace database or *table_name*.
username	The user from which the permissions are revoked. To revoke permissions from multiple users at a time, separate the usernames with commas (,).

The following table lists the permissions that can be revoked.

Permissions

Permission	Description
ALL PRIVILEGES	All permissions except GRANT OPTION.
ALTER	The permission to execute the ALTER TABLE statement.
CREATE	The permission to execute the CREATE TABLE statement.
CREATE USER	The permission to execute the CREATE USER, DROP USER, RENAME USER, and REVOKE ALL PRIVILEGES statements.
CREATE TABLEGROUP	The global permission to execute the CREATE TABLEGROUP statement.
DELETE	The permission to execute the DELETE statement.
DROP	The permission to execute the DROP statement.
GRANT OPTION	The permission to execute the GRANT OPTION statement.
INSERT	The permission to execute the INSERT statement.
SELECT	The permission to execute the SELECT statement.
UPDATE	The permission to execute the UPDATE statement.
SUPER	The permission to execute the SET GLOBAL statement to modify global system parameters.
SHOW DATABASES	The global permission to execute the SHOW DATABASES statement.
INDEX	The permission to execute the CREATE INDEX and DROP INDEX statements.
CREATE VIEW	The permission to create or delete a view.
SHOW VIEW	The permission to execute the SHOW CREATE VIEW statement.
CREATE SYNONYM	The permission to create a synonym.

> **Note**
>
> The permission to execute the CHANGE EFFECTIVE TENANT statement is not limited. Therefore, all users under the system tenant can revoke this permission from other users.

Examples

Run the following command to revoke all permissions from the obsqluser01 user:

```
OceanBase(admin@TEST)>REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'obsqluser01';
Query OK, 0 rows affected (0.03 sec)
```

17.1.4.5.48. SAVEPOINT

Description

You can execute the SAVEPOINT statement to perform a partial rollback of a transaction.

Syntax

1. Create a savepoint:

```
SAVEPOINT spname
```

2. Roll back to a savepoint:

```
ROLLBACK [WORK] to [SAVEPOINT] spname
```

3. Delete a savepoint:

```
RELEASE SAVEPOINT spname
```

Parameters

- spname: specifies the name of the savepoint. Savepoints are unique within a transaction. If the specified name of a savepoint already exists, the new savepoint overwrites the existing savepoint. After a savepoint is created, the transaction can be rolled back to the savepoint. You can also execute the `ROLLBACK` statement to roll back the entire transaction.

Examples

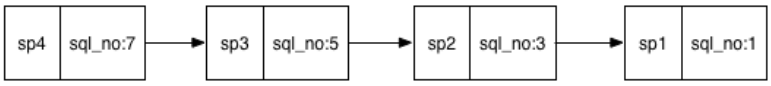
Assume that a transaction has executed the following statements:

sql_no	Statement	Partition
1	update...	p1, p4
	savepoint sp1	
2	update...	p2, p4

sql_no	Statement	Partition
3	update...	p3, p5
	savepoint sp2	
4	update...	p1, p3, p6
5	update...	p1, p5
	savepoint sp3	
6	select...	
7	update...	p5, p6
	savepoint sp4	

Record a savepoint

You can create savepoints before you submit transactions. You must connect the transaction savepoints into a linked list based on the order in which the savepoints are created. The preceding transaction contains seven SQL statements and four savepoints. The following figure shows the linked list of savepoints, where each node records a mapping from the spname to the sql_no.



List of transaction participants

To roll back all changes after an SQL statement is executed in a transaction, the participants and sql_no of each statement must be recorded. The preceding transaction executes seven SQL statements and involves six partitions from p1 to p6.

p1	p2	p3	p4	p5	p6
1,4,5	2	2,3,4	1,2	3,5,7	4,7

Roll back to a savepoint

1. Find the sql_no that corresponds to the spname based on the savepoint linked list.

Assume that you have executed the `ROLLBACK to SAVEPOINT sp2` statement. In the linked list of savepoints, sp2 corresponds to sql_no:3.

2. Find the partitions that correspond to the sql_no based on the list of transaction participants.

The query result shows that the statements whose sql_no values are greater than 3 involve partitions p1, p3, p5, and p6.

3. Roll back the data in partitions

The scheduler initiates a rollback request to the partitions obtained in Step 2. All changes caused by the transaction after sp2 are rolled back in these partitions. Some changes caused by this transaction in p1, p3, and p5 are rolled back, and all changes in p6 are rolled back.

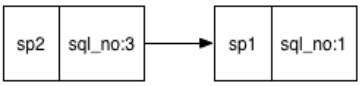
4. Update the transaction participant list

ApsaraDB for OceanBase modifies the transaction participant list and deletes the information about operations that have sql_no greater than 3 from the transaction participant list. p6 is deleted from the transaction participant list because all changes in p6 are rolled back.

p1	p2	p3	p4	p5	p6
1	2	2, 3	1, 2	3	

5. Delete an invalid savepoint

After you execute the `ROLLBACK to SAVEPOINT sp2` statement, the system deletes savepoints sp3 and sp4. In this case, you cannot roll back to sp3 or sp4.



17.1.4.5.49. SCHEMA

You can execute statements that include `SCHEMA` in the same way as you execute the statements that include `DATABASE`. For more information, see the descriptions about the CREATE DATABASE, ALTER DATABASE, and DROP DATABASE statements.

17.1.4.5.50. SELECT

The syntax of SELECT statements is complex. This topic describes the syntax of simple SELECT statements and the SELECT statements that contain set operators.

SIMPLE SELECT

Description

You can execute a simple SELECT statement to query table data.

Syntax

```
simple_select:
SELECT [/*+ hint statement */] [ALL | DISTINCT | UNIQUE | SQL_CALC_FOUND_ROWS]
 select_expr_list FROM from_list [WHERE condition]
  [GROUP BY group_expression_list [WITH ROLLUP] [HAVING condition]]
  [ORDER BY order_expression_list]
  [limit_clause]
  [FOR UPDATE]

select_expr:
 table_name.*
  | table_alias_name.*
  | expr [[AS] column_alias_name]

from_list:
    table_reference [, table_reference ...]

table_reference:
      simple_table
    | joined_table

simple_table:
    table_factor [partition_option] [[AS] table_alias_name]
    | (select_stmt) [AS] table_alias_name
    | (table_reference_list)

joined_table:
      table_reference [INNER] JOIN simple_table [join_condition]
    | table_reference outer_join_type JOIN simple_table join_condition

partition_option:
    PARTITION (partition_name_list)

partition_name_list:
    partition_name [, partition_name ...]

outer_join_type:
    {LEFT | RIGHT | FULL} [OUTER]

join_condition:
    ON expression

condition:
    expression

group_expression_list:
    group_expression [, group_expression ...]

group_expression:
    expression [ASC | DESC]

order_expression_list:
    order_expression [, order_expression ...]

order_expression:
    expression [ASC | DESC]

limit_clause:
  LIMIT {[offset,] row_count |row_count OFFSET offset}
```

Parameters

Parameter	Description
ALL | DISTINCT | UNIQUE | SQL_CALC_FOUND_ROWS	Specifies whether to return distinct table rows. A database table may contain duplicate values. • If you add DISTINCT to the statement, only distinct rows are returned in the query results. • If you add ALL to the statement, all the matched rows are returned. • If you add SQL_CALC_FOUND_ROWS to the statement, only the number of rows is returned. • If you do not add ALL, DISTINCT, UNIQUE, or SQL_CALC_FOUND_ROWS to the statement, the default setting ALL is used.
select_expr	The expressions or column names to query. To specify multiple expressions or column names, separate them with commas (,). You can use an asterisk (*) to indicate that all the columns are queried.
AS othername	Rename the output fields.
FROM table_references	The tables from which data is queried. You can query data from multiple tables.
WHERE where_conditions	The filter conditions. The query results contain the data that can satisfy the filter conditions. This clause is optional. where_conditions specifies an expression.
GROUP BY group_by_list	Divide data into groups.
HAVING search_confitions	The filter conditions. HAVING clauses are similar to WHERE clauses. The difference between HAVING and WHERE clauses is that you can use aggregate functions in HAVING clauses, such as SUM and AVG.
ORDER BY order_list order_list : colname [ASC | DESC] [,colname [ASC | DESC]...]	Display the query results in ascending or descending order. ASC indicates the ascending order and DESC indicates the descending order. If you do not specify the order, the default order ASC is used.

Parameter	Description
[LIMIT {[offset,] row_count |row_count OFFSET offset}]	Limit the number of rows that are returned by the SELECT statement. You can specify one or two arguments of the numeric data type for the LIMIT clause. The arguments must be integer constants. • If you specify two arguments, the first argument specifies the offset for the first row to be returned and the second argument specifies the maximum number of rows to be returned. The initial offset for the first row is 0 instead of 1. • If you specify only one argument, the argument specifies the maximum number of rows to be returned and the offset is 0.
FOR UPDATE	Apply an exclusive lock on each row of the query results. This prevents other transactions from concurrently updating the rows. This also prevents other transactions from concurrently reading the rows for which some transaction isolation levels are specified.
PARTITION(partition_list)	The partition information of the specified tables. For example, partition(p0,p1...).

Examples

The following examples are based on table a.

Table a

id	name	num
1	a	100
2	b	200
3	a	50

- Query the name column from table a.

```
SELECT name FROM a;
```

```
+------+
| name |
+------+
| a    |
| b    |
| a    |
+------+
3 rows in set (0.01 sec)
```

- Return the distinct rows for the name column.

```
SELECT DISTINCT name FROM a;
```

```
+------+
| name |
+------+
| a    |
| b    |
+------+
2 rows in set (0.01 sec)
```

- In table a, query the id, name, and num columns, divide the num column values by 2, and return the calculation result of the num column in a column named avg.

```
SELECT id, name, num/2 AS avg FROM a;
```

```
+------+------+------+
| id   | name | avg  |
+------+------+------+
|    1 | a    |   50 |
|    2 | b    |  100 |
|    3 | a    |   25 |
+------+------+------+
3 rows in set (0.00 sec)
```

- In table a, find the 'a' value for the name column, and return the values for the id, name, and num columns.

```
SELECT id, name, num FROM a WHERE name = 'a';
```

```
+------+------+------+
| id   | name | num  |
+------+------+------+
|    1 | a    |  100 |
|    3 | a    |   50 |
+------+------+------+
2 rows in set (0.01 sec)
```

- In table a, query the id and name columns, group the num column values by name, and return the sum of the num column values.

```
SELECT id, name, SUM(num) FROM a GROUP BY name;
```

```
+------+------+----------+
| id   | name | SUM(num) |
+------+------+----------+
|    1 | a    |      150 |
|    2 | b    |      200 |
+------+------+----------+
2 rows in set (0.00 sec)
```

- In table a, query the id and name columns, group the num column values by name, and calculate the sum of the num column values. Then, return the sum value that is less than 160.

```
SELECT id, name, SUM(num) as sum FROM a GROUP BY name HAVING SUM(num) < 160;
```

```
+------+------+------+
| id   | name | sum  |
+------+------+------+
|    1 | a    |  150 |
+------+------+------+
1 row in set (0.01 sec)
```

- In table a, query the id, name, and num columns, and sort and return the result set by the num column in ascending order.

```
SELECT * FROM a ORDER BY num ASC;
```

```
+------+------+------+
| id   | name | num  |
+------+------+------+
|    3 | a    |   50 |
|    1 | a    |  100 |
|    2 | b    |  200 |
+------+------+------+
3 rows in set (0.00 sec)
```

- In table a, query the id, name, and num columns, and sort and return the result set by the num column in descending order.

```
SELECT * FROM a ORDER BY num DESC;
```

- In table a, query the id, name, and num columns, and use the LIMIT clause to return two rows that start from the second row.

```
SELECT * FROM a LIMIT 1,2;
```

```
+------+------+------+
| id   | name | num  |
+------+------+------+
|    2 | b    |  200 |
|    3 | a    |   50 |
+------+------+------+
2 rows in set (0.00 sec)
```

SELECT statements that contain set operators

Description

You can execute the SELECT statement that contains the UNION, MINUS, or INTERSECT operator to combine query results.

Syntax

```
select_clause_set:
    simple_select [ UNION | UNION ALL | EXCEPT | INTERSECT] select_clause_set_left
  [ORDER BY sort_list_columns] [limit_clause]
select_clause_set_right:
    simple_select |
  select_caluse_set
```

Parameters

Parameter	Description
UNION ALL	Combine the results of two queries and return all rows.
UNION	Combine the results of two queries and return distinct rows.
EXCEPT	Return the distinct rows in the left query that are not selected by the right query.
INTERSECT	Return the distinct rows that are selected by the left and right queries.

Examples

The following examples are based on tables t1 and t2.

```
create table t1 (c1 int, c2 int);
create table t2 (c1 int, c2 int);
insert into t1 values (1, -1), (2, -2);
insert into t2 values (1, 1), (2, -2), (3, 3);
```

- Query all rows from t1 and t2.

```
SELECT C1, C2 FROM T1 UNION ALL SELECT C1, C2 FROM T2;
+------+------+
| C1   | C2   |
+------+------+
|    1 |   -1 |
|    2 |   -2 |
|    1 |    1 |
|    2 |   -2 |
|    3 |    3 |
+------+------+
```

- Query all distinct rows from t1 and t2.

```
SELECT C1, C2 FROM T1 UNION SELECT C1, C2 FROM T2;
+------+------+
| C1   | C2   |
+------+------+
|    1 |   -1 |
|    2 |   -2 |
|    1 |    1 |
|    3 |    3 |
+------+------+
```

- Query the rows that exist in t1 and t2.

```
SELECT C1, C2 FROM T1 INTERSECT SELECT C1, C2 FROM T2;
+------+------+
| C1   | C2   |
+------+------+
|    2 |   -2 |
+------+------+
```

- Query the rows that exist in t1 but do not exist in t2.

```
SELECT C1, C2 FROM T1 EXCEPT SELECT C1, C2 FROM T2;
+------+------+
| C1   | C2   |
+------+------+
|    1 |   -1 |
+------+------+
```

- Query all the distinct rows from t1 and t2, and return the first two rows ordered by the c2 column in descending order.

```
SELECT C1, C2 FROM T1 UNION SELECT C1, C2 FROM T2 ORDER BY C2 DESC LIMIT 2;
+------+------+
| C1   | C2   |
+------+------+
|    3 |    3 |
|    1 |    1 |
+------+------+
```

17.1.4.5.51. SESSION

Description

You can execute the SESSION statement to close a session.

Syntax

```
session_stmt:
    KILL [CONNECTION] session_id;

session_id:
    INT_VALUE
```

Parameters

Parameter	Description
session_id	The ID of the session that you want to close.

Examples

- Close sessions 3221502221 and 3221750376.

```
OceanBase(admin@test)>show processlist;
+------------+-------+--------------------+------+---------+------+--------+-----------------------
----------+
| Id         | User  | Host               | db   | Command | Time | State  | Info
|
+------------+-------+--------------------+------+---------+------+--------+-----------------------
----------+
| 3221502221 | admin | 100.xx.xx.xx:44775 | test | Sleep   |   10 | SLEEP  | NULL
|
| 3221502083 | admin | 100.xx.xx.xx:44720 | test | Query   |    0 | ACTIVE | show processlist
|
| 3221502317 | NULL  | 100.xx.xx.xx:41414 | test | Query   |    0 | ACTIVE | REMOTE/DISTRIBUTE PLAN
EXECUTING |
| 3221750377 | NULL  | 100.xx.xx.xx:41414 | test | Query   |    0 | ACTIVE | REMOTE/DISTRIBUTE PLAN
EXECUTING |
| 3221750376 | admin | 100.xx.xx.xx:43783 | test | Sleep   |    4 | SLEEP  | NULL
|
+------------+-------+--------------------+------+---------+------+--------+-----------------------
----------+
5 rows in set (0.04 sec)

OceanBase(admin@test)>kill 3221502221;
Query OK, 0 rows affected (0.00 sec)

OceanBase(admin@test)>kill connection 3221750376;
Query OK, 0 rows affected (0.05 sec)
```

17.1.4.5.52. SET PASSWORD

Description

You can execute the SET PASSWORD statement to change the password for the current user or another user.

Syntax

```
SET PASSWORD [FOR user] = password_option;

password_option: {
PASSWORD('authstring')
|'hashstring'
}
```

Parameters

Parameter	Description
FOR user	If you do not use the FOR user clause, the statement changes the password for the current user. After a user logs on to the ApsaraDB for OceanBase system, the user can change its own password. If you use the FOR user clause, the statement changes the password for a specified user. To change the password for a specified user, ensure that you have the global permission to execute the CREATE USER statement.

Examples

Run the following command to change the password of the sqluser01 user to abc123:

```
oceanBase(admin@TEST)>set password for sqluser01 = password('abc123');
Query OK, 0 rows affected (0.02 sec)
```

17.1.4.5.53. SHOW GRANTS

Description

You can execute the SHOW GRANTS statement to view user permissions as a system administrator.

Syntax

```
SHOW GRANTS [FOR username];
```

Parameters

Parameter	Description
FOR username	If you do not specify the username, this statement returns the permissions that have been granted to the current user. You can view the permissions of the current user. To view the permissions of another specified user, ensure you have the SELECT permission on the mysql.user table.

Examples

View the permissions of the obsqluser01 user.

```
oceanBase(admin@TEST)>show grants for obsqluser01;
+-------------------------------------+
| Grants for obsqluser01@%            |
+-------------------------------------+
| GRANT USAGE ON *. * TO 'obsqluser01' |
+-------------------------------------+
1 row in set (0.02 sec)
```

17.1.4.5.54. SHOW RECYCLEBIN

Description

You can execute the SHOW RECYCLEBIN statement to view objects in the recycle bin.

Syntax

```
 SHOW RECYCLEBIN;
```

Parameters

None.

Examples

- View objects in the recycle bin.

```
OceanBase(admin@test)> create table t1(c1 int);
Query OK, 0 rows affected (0.24 sec)

OceanBase(admin@test)> drop table t1;
Query OK, 0 rows affected (0.07 sec)

OceanBase(admin@test)> show recyclebin;
+-------------------------------------------+---------------+-------+----------------------------+
| OBJECT_NAME                               | ORIGINAL_NAME | TYPE  | CREATETIME                 |
+-------------------------------------------+---------------+-------+----------------------------+
| __recycle_$_1_1099511628776_1099511677777 | t1            | TABLE | 2017-10-20 17:27:40.881506 |
+-------------------------------------------+---------------+-------+----------------------------+
1 row in set (0.02 sec)
```

17.1.4.5.55. TRANSACTION

Description

You can execute the TRANSACTION statement to start a transaction.

A database transaction is a single logical unit of work that consists of a collection of operations. Transaction processing ensures that SQL operations in a batch are all executed or are not executed at all. You can use transactions to maintain the data integrity of databases.

An explicit transaction is a user-defined or user-specified transaction. An explicit transaction is a transaction that starts with the BEGIN TRANSACTION, BEGIN, or BEGIN WORK statement and ends with the COMMIT or ROLLBACK statement. BEGIN and BEGIN WORK are supported as aliases of the START TRANSACTION statement.

Syntax

```
transaction_stmt:
      START TRANSACTION [READ ONLY | READ WRITE];
    | BEGIN [WORK];
    | COMMIT [WORK];
    | ROLLBACK [WORK];
    | SET TRANSACTION {READ ONLY | READ WRITE};
```

Parameters

Parameter	Description
START TRANSACTION [READ ONLY | READ WRITE]	Start a transaction. After a transaction is started, the SQL statements that follow the START TRANSACTION statement, such as INSERT, UPDATE, and DELETE, take effect only when the transaction is explicitly committed. The READ ONLY clause specifies that the transaction is started in read-only mode. This prevents you from modifying the transaction after the transaction is started. The READ WRITE clause specifies that the transaction is started in read/write mode. By default, the read/write mode is used.

Parameter	Description
BEGIN	BEGIN and BEGIN WORK are supported as aliases of the START TRANSACTION statement.
COMMIT	Commit the current transaction.
ROLLBACK	Roll back the current transaction.
SET TRANSACTION {READ ONLY | READ WRITE}	Set the current transaction mode. Valid values: READ ONLY and READ WRITE.

Examples

Assume that table a is created, as shown in the following table.

id	name	num	sell_date
1	a	100	2013-06-21 10:06:43
2	b	200	2013-06-21 13:07:21
3	a	50	2013-06-21 13:08:15

1. Run the following commands in sequence to start a transaction. The transaction completes the following operations: Find the row where the value for the id column is 3, change the value for the name column to c, and then insert a row that contains a sale record of product a.

```
OceanBase(admin@test)> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

OceanBase(admin@test)> UPDATE a SET name = 'c' WHERE id = 3;
Query OK, 1 rows affected (0.00 sec)

OceanBase(admin@test)> INSERT INTO a VALUES (4, 'a', 30, '2013-06-21 16:09:13');
Query OK, 1 rows affected (0.00 sec)

OceanBase(admin@test)> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

2. After you commit the transaction, run the following command to view table a:

```
SELECT * FROM a;
```

Return result:

id	name	num	sell_date
1	a	100	2013-06-21 10:06:43
2	b	200	2013-06-21 13:07:21
3	c	50	2013-06-21 13:08:15
4	a	30	2013-06-21 16:09:13

> **Notice**
>
> Before you commit a transaction, you can check whether the operations in the transaction have taken effect. For example, you can insert **SELECT * FROM a;** before you execute the COMMIT statement. The session within which this transaction is executed can read the updated result. A session outside this transaction cannot read the updated result. Before the transaction is committed, your previous operations are invisible outside the transaction session. To roll back a transaction, replace COMMIT with ROLLBACK.

17.1.4.5.56. TRUNCATE TABLE

Description

You can execute the TRUNCATE TABLE statement to delete all data in a specified table and retain the table schema, including the partition information of the table. The TRUNCATE TABLE statement implements the same logic as the DELETE FROM statement. You can use DELETE FROM to delete all rows. To execute the TRUNCATE TABLE statement, ensure that you have the permissions to create and delete tables. The TRUNCATE TABLE statement is a Data Definition Language (DDL) statement.

The TRUNCATE TABLE and DELETE FROM statements have the following differences:

- The TRUNCATE TABLE statement deletes and creates the table again, whereas the DELETE FROM statement deletes rows one after another. Therefore, The response time for the TRUNCATE TABLE statement is shorter than that for the DELETE FROM statement.
- The output of the TRUNCATE TABLE statement shows that the number of affected rows is always 0.
- If you execute the TRUNCATE TABLE statement, each auto-incremented value is reset to the start value. The table manager does not store the latest auto-incremented value.
- You cannot execute the TRUNCATE TABLE statement when a transaction is being processed or the table is locked. If you execute the statement in these scenarios, the system returns errors.
- If the file that defines the table is valid, you can execute the TRUNCATE TABLE statement to create the table again as an empty table. This applies even if the data or the index file is corrupted.

Syntax

```
TRUNCATE [TABLE] table_name;
```

Parameters

Parameter	Description
table_name	The name of the table.

Examples

Remove all records from the tb1 table.

```
TRUNCATE [TABLE] tb1;
```

17.1.4.5.57. UPDATE

Description

You can execute the UPDATE statement to change field values in a table.

Syntax

```
UPDATE [IGNORE] table_references
    SET update_asgn_list
    [WHERE where_condition]
    [ORDER BY order_list]
    [LIMIT row_count];

table_references:
    tbl_name [PARTITION (partition_name,...)] [, ...]

update_asgn_list:
    column_name = expr [, ...]

order_list:
    column_name [ASC|DESC] [, column_name [ASC|DESC]…]
```

Parameters

Parameter	Description
IGNORE	Ignore the errors that occur when the INSERT statement is being executed.
table_references	The name of the table that you want to update. To update multiple tables at a time, separate the table names with commas (,).
where_condition	The filter conditions.
row_count	The maximum number of rows that can be updated.

Parameter	Description
tbl_name	The name of the table.
partition_name	The name of the partition.
column_name	The name of the column.
column_name ASC	Sort the table by values in a specified column in ascending order and then update the table.
column_name DESC	Sort the table by values in a specified column in descending order and then update the table.

Notes

You cannot execute the UPDATE statement on the output value of a subquery. This rule applies regardless of the number of tables that you want to update. For example, this statement is not allowed:

`update (select * from t1) set c1 = 100;` .

Examples

1. Create sample tables t1 and t2.

```
OceanBase(admin@test)>create table t1(c1 int primary key, c2 int);
Query OK, 0 rows affected (0.16 sec)
OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |    1 |
|  2 |    2 |
|  3 |    3 |
|  4 |    4 |
+----+------+
4 rows in set (0.06 sec)

OceanBase(admin@test)>create table t2(c1 int primary key, c2 int) partition by key(c1) partitions 4;
Query OK, 0 rows affected (0.19 sec)
OceanBase(admin@test)>select * from t2;
+----+------+
| c1 | c2   |
+----+------+
|  5 |    5 |
|  1 |    1 |
|  2 |    2 |
|  3 |    3 |
+----+------+
4 rows in set (0.02 sec)
```

2. In the t1 table, find the row that matches the t1.c1 = 1 condition, and change the value at the intersection of this row and the c2 column to 100.

```
OceanBase(admin@test)>update t1 set t1.c2 = 100 where t1.c1 = 1;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |  100 |
|  2 |    2 |
|  3 |    3 |
|  4 |    4 |
+----+------+
4 rows in set (0.01 sec)
```

3. Sort the t1 table by the values in column c2, and change the first two values in column c2 to 100.

```
OceanBase(admin@test)>update t1 set t1.c2 = 100 order by c2 limit 2;
Query OK, 2 rows affected (0.02 sec)
Rows matched: 2  Changed: 2  Warnings: 0

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |  100 |
|  2 |  100 |
|  3 |    3 |
|  4 |    4 |
+----+------+
4 rows in set (0.01 sec)
```

4. In the p2 partition of the t2 table, find the row that matches the t2.c1 > 2 condition. Then, change the value at the intersection of this row and the c2 column to 100.

```
OceanBase(admin@test)>update t2 partition(p2) set t2.c2 = 100 where t2.c1 > 2;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

OceanBase(admin@test)>select * from t2;
+----+------+
| c1 | c2   |
+----+------+
|  5 |    5 |
|  1 |    1 |
|  2 |    2 |
|  3 |  100 |
+----+------+
4 rows in set (0.06 sec)
```

5. Update multiple tables. In the t1 and t2 tables, find the rows that match the t1.c1 = t2.c1 condition. Change the value at the intersection of the matching row in the t1 table and the c2 column to 100. Change the value at the intersection of the matching row in the t2 table and the c2 column to 200.

```
OceanBase(admin@test)>update t1,t2 set t1.c2 = 100, t2.c2 = 200 where t1.c2 = t2.c2;
Query OK, 6 rows affected (0.03 sec)
Rows matched: 6  Changed: 6  Warnings: 0

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |  100 |
|  2 |  100 |
|  3 |  100 |
|  4 |    4 |
+----+------+
4 rows in set (0.00 sec)

OceanBase(admin@test)>select * from t2;
+----+------+
| c1 | c2   |
+----+------+
|  5 |    5 |
|  1 |  200 |
|  2 |  200 |
|  3 |  200 |
+----+------+
4 rows in set (0.01 sec)
```

6. Update multiple tables. In the p2 partition in the t1 and t2 tables, find the rows that match the t1.c1 = t2.c1 condition. Change the value at the intersection of the matching row in the t1 table and the c2 column to 100. Change the value at the intersection of the matching row in the t2 table and the c2 column to 200.

```
OceanBase(admin@test)>update t1,t2 partition(p2) set t1.c2 = 100, t2.c2 = 200 where t1.c2 = t2.c2;
Query OK, 6 rows affected (0.02 sec)
Rows matched: 6  Changed: 6  Warnings: 0

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |  100 |
|  2 |  100 |
|  3 |  100 |
|  4 |    4 |
+----+------+
4 rows in set (0.01 sec)

OceanBase(admin@test)>select * from t2;
+----+------+
| c1 | c2   |
+----+------+
|  5 |    5 |
|  1 |  200 |
|  2 |  200 |
|  3 |  200 |
+----+------+
4 rows in set (0.01 sec)
```

7. Update the values in updatable view v.

```
OceanBase(admin@test)>create view v as select * from t1;
Query OK, 0 rows affected (0.07 sec)

OceanBase(admin@test)>update v set v.c2 = 100 where v.c1 = 1;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

OceanBase(admin@test)>select * from v;
+----+------+
| c1 | c2   |
+----+------+
|  1 |  100 |
|  2 |    2 |
|  3 |    3 |
|  4 |    4 |
+----+------+
4 rows in set (0.01 sec)
```

17.1.5. SQL Reference (Oracle Mode)

17.1.5.1. Compatibility with Oracle

ApsaraDB for OceanBase supports most of the basic SQL syntax in Oracle. This indicates that when your business is transferred from Oracle to ApsaraDB for OceanBase, you do not need to spend a large amount of time learning new syntax. In addition, your business can be smoothly migrated from Oracle to ApsaraDB for OceanBase.

For optimization and development reasons, ApsaraDB for OceanBase does not support some features or the effect of some features in ApsaraDB for OceanBase is different from that in Oracle. This document describes the compatibility between ApsaraDB for OceanBase and Oracle by topic.

SQL syntax

- ApsaraDB for OceanBase supports the basic SQL syntax in Oracle.
- If some features are unavailable, the system reports errors that indicate that the syntax is not supported. For example, when hierarchical queries include multi-table joins, the system reports the errors.
- ApsaraDB for OceanBase does not support some complex online analytical processing (OLAP) syntax of Oracle, such as pattern matching, `PIVOT`, and `UNPIVOT` functions, polymorphic table functions, and frequent itemset computing.

SQL data types

- Oracle supports 24 data types. ApsaraDB for OceanBase supports 18 data types. For more information, see Built-in data types.
- For the optimization reason, the `LONG` and `LONG RAW` data types are obsolete. Therefore, ApsaraDB for OceanBase does not support these two data types.
- In ApsaraDB for OceanBase, a large object data type stores a maximum of 48 MB data and the performance is low. Therefore, we recommend that you do not use large object data types in complex scenarios. For more information, see Large object data types.

Character sets and collations

- ApsaraDB for OceanBase supports the UTF-8, GBK, GB18030, and national character set.
- ApsaraDB for OceanBase supports only case-sensitive collations.

- ApsaraDB for OceanBase does not support the collations of multilingual semantics.

Built-in functions

Oracle supports 117 built-in functions. ApsaraDB for OceanBase supports 103 built-in functions. For more information, see Functions.

System views

- Oracle supports more than 400 dictionary views. ApsaraDB for OceanBase is compatible with 17 of the 400 dictionary views.
  - ALL_CONS_COLUMNS
  - ALL_CONSTRAINTS
  - ALL_IND_COLUMNS
  - ALL_INDEX
  - ALL_OBJECTS
  - ALL_PART_KEY_COLUMNS
  - ALL_PART_TABLES
  - ALL_SEQUENCE
  - ALL_SOURCE
  - ALL_SUBPART_KEY_COLUMNS
  - ALL_SYNONYMS
  - ALL_TAB_COLUMNS
  - ALL_TAB_PARTITIONS
  - ALL_TABLES
  - ALL_TYPES
  - ALL_USERS
  - ALL_VIEWS
- Oracle supports more than 700 performance views. ApsaraDB for OceanBase is compatible with 3 of the 700 performance views.
  - V$SYSTEM_EVENT
  - V$SESSION_WAIT
  - V$NLS_PARAMETERS

SQL features

ApsaraDB for OceanBase supports the following core SQL features in Oracle:

- Plan cache
- Outline binding
- Plan management evolution
- Cost-based optimizer
- Cost-based rewriting
- Prepared statements
- Global indexes
- Function-based indexes

Foreign keys

- ApsaraDB for OceanBase supports foreign keys.
- In ApsaraDB for OceanBase, you cannot add the following foreign key constraints: `DISABLE` and `ENABLE`.
- In ApsaraDB for OceanBase, you cannot add foreign key constraints in `ALTER TABLR` statements.
- ApsaraDB for OceanBase does not support `SET NULL` in cascade clauses.

Triggers

- ApsaraDB for OceanBase supports only row-level triggers.
- In ApsaraDB for OceanBase, you can create triggers on only tables. You cannot create triggers on views.
- In ApsaraDB for OceanBase, you cannot perform the `DISABLE` or `ENABLE` operation on triggers.

Database links

- ApsaraDB for OceanBase does not support database links.

Synonyms

In ApsaraDB for OceanBase, you can create synonyms for objects, such as tables, views, synonyms, and sequences. You can also create public synonyms.

Updatable views

ApsaraDB for OceanBase does not support the `WITH CHECK OPTION` clauses.

Constraints

- ApsaraDB for OceanBase supports the `CHECK`, `UNIQUE`, and `NOT NULL` constraints.
- ApsaraDB for OceanBase does not support the `DISABLE` operation on the `UNIQUE` constraint.

Hint

Oracle has 73 hints. ApsaraDB for OceanBase is compatible with 25 of the 73 hints. In addition, the number of hints that are specific to ApsaraDB for OceanBase is 20.

Oracle hint with which ApsaraDB for OceanBase is compatible	Hint specific to ApsaraDB for OceanBase
USE_BNL	INDEX Hint
NO_USE_BNL Hint	FULL Hint
USE_PX Hint	LEADING Hint
NO_USE_PX Hint	ORDERED Hint

Oracle hint with which ApsaraDB for OceanBase is compatible	Hint specific to ApsaraDB for OceanBase
USE_JIT Hint	USE_MERGE Hint
NO_USE_JIT Hint	NO_USE_MERGE Hint
USE_HASH_AGGREGATION Hint	USE_HASH Hint
NO_USE_HASH_AGGREGATION Hint	NO_USE_HASH Hint
USE_LATE_MATERIALIZATION Hint	USE_NL Hint
NO_USE_LATE_MATERIALIZATION Hint	NO_USE_NL Hint
USE_NL_MATERIALIZATION Hint	PARALLEL Hint
NO_USE_NL_MATERIALIZATION Hint	PQ_DISTRIBUTE Hint
PLACE_GROUP_BY Hint	NO_REWRITE Hint
NO_PLACE_GROUP_BY Hint	NO_EXPAND Hint
NO_PRED_DEDUCE Hint	USE_CONCAT Hint
READ_CONSISTENCY Hint	MERGE Hint
FROZEN_VERSION Hint	NO_MERGE Hint
QUERY_TIMEOUT Hint	UNNEST Hint
LOG_LEVEL Hint	NO_UNNEST Hint
USE_PLAN_CACHE Hint	QB_NAME Hint
TRANS_PARAM Hint	
TRACING Hint	
STAT Hint TOPK Hint	

Oracle hint with which ApsaraDB for OceanBase is compatible	Hint specific to ApsaraDB for OceanBase
TRACE_LOG Hint	

Security-related items

TDE

ApsaraDB for OceanBase supports the transparent data encryption (TDE) feature that is compatible with Oracle and does not allow you to encrypt redo log files.

Auditing

- ApsaraDB for OceanBase supports standard auditing of Oracle but does not support unified auditing.
- You can store the audit results of ApsaraDB for OceanBase in files or internal audit tables.
- ApsaraDB for OceanBase supports two types of auditing: statement auditing and object auditing. Object auditing supports only tables, sequences, and package objects.
- ApsaraDB for OceanBase does not support network auditing or Fine Grained Auditing (FGA).

Label Security

- ApsaraDB for OceanBase supports the Label Security feature of Oracle.
- In ApsaraDB for OceanBase, you cannot perform a `DISABLE` operation on a created policy that has taken effect or perform an `ENABLE` operation on this policy after a `DISABLE` operation.

SSL encryption for transmission links

ApsaraDB for OceanBase allows you to encrypt transmission links between clients and ApsaraDB for OceanBase servers.

Partitioning support

ApsaraDB for OceanBase supports partitioning and subpartitioning and the following partitioning methods: hash partitioning, range partitioning, and list partitioning.

The following table lists the details about the support for subpartitioning.

Partition or subpartition	Hash	Range	List
Hash	Not supported	Supported	Supported
Range	Supported	Not supported	Supported
List	Supported	Supported	Not supported

- ApsaraDB for OceanBase supports only basic operation commands for partition maintenance, such as the `ADD PARTITION` operation for partitions.

- ApsaraDB for OceanBase does not support complex operations for partition maintenance, such as `SPLIT`, `MERGE`, and `EXCHANGE` operations on partitions.
- ApsaraDB for OceanBase does not support the `TRUNCATE` operation on partitions.
- ApsaraDB for OceanBase supports only homogeneous subpartitioning and does not support heterogeneous subpartitioning.

Parallel queries

- ApsaraDB for OceanBase supports parallel queries that are similar to those of Oracle. In ApsaraDB for OceanBase, you need to manually specify the degree of parallelism (DOP). ApsaraDB for OceanBase does not support the Auto DOP feature.
- ApsaraDB for OceanBase does not support Product Data Markup Language (PDML).

17.1.5.2. SQL overview

Structured Query Language (SQL) is a programming language that is used for specific purposes. All the programs and users can use SQL to access data in ApsaraDB for OceanBase in the same way as access to other current popular relational databases. Even if some platforms and tools allow users to directly access the database through interfaces or UIs, the underlying layers of these platforms and tools still use SQL to access databases.

SQL history

In June 1970, Dr. E. F. Codd in the laboratory of IBM in San Jose, California published the paper A Relational Model of Data for Large Shared Data Banks in the Association for Computing Machinery (ACM) journal. He proposed the concept of the relational model.

In 1974, D.D.Chamberlin and R.F.Boyce in the same laboratory developed a set of Structured English Query Language (SEQUEL) in the relational database system SystemR that was developed by IBM. In IBM Journal of R&D of November 1976, they published a new version of SQL that was named SEQUEL/2. In 1980, SEQUEL/2 was renamed as SQL.

In 1979, Oracle first provided SQL for commercial use and IBM also implemented SQL in the DB2 and SQL/DS database systems.

Nowadays, SQL has become the standard language for relational database management systems (RDBMSs).

SQL standards

In October 1986, the American National Standards Institute (ANSI) adopted SQL as the standard language for RDBMSs and named the language ANSI X3. 135-1986. The International Organization for Standardization (ISO) also adopted SQL as the international standard.

In 1989, ANSI adopted the SQL standard language that was defined in the ANSI X3.135-1989 report, and named the language ANSI SQL 89. This standard replaced the previous version ANSI X3.135-1986.

The following list provides a brief history for SQL development:

- 1986, ANSI X3.135-1986, ISO/IEC 9075:1986, SQL-86
- 1989, ANSI X3.135-1989, ISO/IEC 9075:1989, SQL-89
- 1992, ANSI X3.135-1992, ISO/IEC 9075:1992, SQL-92 (SQL2)
- 1999, ISO/IEC 9075:1999, SQL:1999 (SQL3)
- 2003, ISO/IEC 9075:2003, SQL:2003
- 2008, ISO/IEC 9075:2008, SQL:2008
- 2011, ISO/IEC 9075:2011, SQL:2011

Nowadays, the content in most of mentioned SQL standards is actually the basic and core part of SQL-92. ApsaraDB for OceanBase also complies with the SQL-92 standard.

Running of SQL

SQL is an interface that is used to access a relational database, such as the interfaces of ApsaraDB for OceanBase an Oracle, and MySQL. All the SQL statements are the instructions for databases.

Generally, SQL includes the following five parts:

1. Data Query Language (DQL): also known as the data retrieval language. It is used to retrieve data from tables and describe how to return the data to programs for output. DQL does not change the data content that is stored in databases.
2. Data Manipulation Language (DML): changes the data content that is stored in databases. This indicates that the language adds, modifies, and deletes data.
3. Transaction Control Language (TCL): ensures the integrity and consistency of databases. The DML statements in the same transaction must be all successful or fail.
4. Data Control Language (DCL): the commands that control the permissions to access data. This language controls the permission of a specified account to access the specified database resources.
5. Data Definition Language (DDL): defines, modifies, and deletes the database resources, such as creating and deleting tables.

Portability of SQL

SQL is a standard language for accessing databases. All the major relational databases support SQL. Therefore, all the programs that are written in SQL are portable. Typically, you can migrate a program from one relational database to the other relational database only after you make only a few modifications.

Vocabulary conventions

- **Bold** indicates the graphical user interface (GUI) elements associated with operations or the terms that are defined in text or a vocabulary.
- Reserved words, keywords, identifiers, and parameters are not case-sensitive. These items are written in uppercase to facilitate reading and identification.
- The ways in which SQL statements are terminated vary based on the programming environments. In this document, a semicolon (;) is used to identify the end of an SQL statement.
- `Inline code` indicates the code that is referenced in the document.
- To highlight the important information, the text, such as description, notes, and importance, are written in bold in this document.
- The text of optional parameters in this document is enclosed in square brackets, such as [-n, -quiet].

17.1.5.3. Pseudocolumns

Pseudocolumns behave the same as table columns, but do not store values. Therefore, pseudocolumns have only the read attribute. You cannot perform operations on pseudocolumns, such as inserting, updating, and deleting data.

> **Notice**
>
> ApsaraDB for OceanBase does not support the `ROWID` pseudocolumn.

Hierarchical query pseudocolumns

Hierarchical query pseudocolumns are valid only in hierarchical queries. To define hierarchical relationships in queries, you must use a `CONNECT BY` clause.

CONNECT_BY_ISCYCLE pseudocolumn

The `CONNECT_BY_ISCYCLE` pseudocolumn is used to assist in marking the row from which a loop starts. If the child node of the current row is also one of the ancestor nodes of the current row, `CONNECT_BY_ISCYCLE` returns 1. Otherwise, 0 is returned.

`CONNECT_BY_ISCYCLE` must be used in conjunction with `NOCYCLE` of the `CONNECT BY` clause. Otherwise, an error is reported for the query result because a loop exists in the tree structure result.

CONNECT_BY_ISLEAF pseudocolumn

The `CONNECT_BY_ISLEAF` pseudocolumn is used to assist in marking leaf nodes in a hierarchy structure. If the current row does not have a child node, that is, the current row is the leaf node of the tree, 1 is returned. Otherwise, 0 is returned.

LEVEL pseudocolumn

The `LEVEL` pseudocolumn is used to assist in marking the hierarchy of nodes. In the hierarchy structure, the root is at level 1, the child nodes of the root is at level 2, and so on. For example, the return value of `LEVEL` for the root node is 1, the return value of `LEVEL` for the child node of the root node is 2, and so on.

Sequence pseudocolumns

A sequence pseudocolumn is an auto-increment numeric sequence that is generated by the database based on specific rules. The sequence pseudocolumn is often used as primary keys and unique keys due to its auto-increment characteristics. You can use the following two methods to obtain the values of the sequence pseudocolumn:

- `CURRVAL`: returns the current value of the sequence.
- `NEXTVAL`: returns the next auto-increment value of the sequence.

When you use a sequence pseudocolumn, you must add the sequence name before `CURRVAL` or `NEXTVAL` and use a period (.) for reference. For example, if the name of a sequence is **SEQ_FOO**, you can use `SEQ_FOO.CURRVAL` to obtain the current value of the **SEQ_FOO** sequence. You can also use `SEQ_FOO.NEXTVAL` to obtain the next auto-increment value of the **SEQ_FOO** sequence.

Application scenarios of sequence pseudocolumns

You can use the `CURRVAL` and `NEXTVAL` values of the sequence pseudocolumns in the following positions:

- The select lists of `SELECT` statements that are in subqueries, materialized views, or views
- The select lists of subqueries in `INSERT` statements
- The `VALUES` clauses in `INSERT` statements
- The `SET` clauses in `UPDATE` statements

You cannot use the `CURRVAL` and `NEXTVAL` values of the sequence pseudocolumns in the following positions:

- The subqueries in `DELETE`, `SELECT`, or `UPDATE` statements
- The queries of views or materialized views
- The `SELECT` statements that contain the `DISTINCT` operator
- The `SELECT` statements that contain the `GROUP BY` or `ORDER BY` clause
- The `SELECT` statement that is combined with another `SELECT` statement by using the `UNION`, `INTERSECT`, or `MINUS` set operator
- The `WHERE` clauses in `SELECT` statements
- The `DEFAULT` values of columns in `CREATE TABLE` or `ALTER TABLE` statements
- The conditions of `CHECK` constraints

How to use sequence pseudocolumns

When you create a sequence, you must specify its initial value and step size. When you reference `NEXTVAL` for the first time, the initial value of the sequence is returned. When you subsequently reference `NEXTVAL`, a new value is returned after the return value of the previous sequence plus the step size that is specified for the sequence. When you reference `CURRVAL` at any time, the current value of the sequence, that is, the return value for the last reference to `NEXTVAL`, is returned.

Before you reference the `CURRVAL` pseudocolumn of a sequence in a session, you must first initialize the sequence value for the current session by using the `NEXTVAL` pseudocolumn of the sequence.

When you create a sequence, you can define its initial value and the increment between values of the sequence. When you reference `NEXTVAL` for the first time, the initial value of the sequence is returned. When you subsequently reference `NEXTVAL`, the sequence value is incremented by the defined increment and a new value is returned. For any reference to `CURRVAL`, the current value of the sequence, that is, the return value for the last reference to `NEXTVAL`, is always returned. For more information about how to create sequences, see CREATE SEQUENCE.

If you reference `NEXTVAL` in a single SQL statement, ApsaraDB for OceanBase increments the sequence in the following ways:

- Each time the outer query block of a `SELECT` statement returns one row, the sequence is incremented once. Such a query block can appear in the following positions:
  - Top-level `SELECT` statements.
  - `INSERT... SELECT` statements. If data is inserted into multiple tables, `NEXTVAL` must be placed in the `VALUES` clause. Each time the subquery returns one row, the sequence is incremented once, even if multiple branches reference `NEXTVAL`.
  - `CREATE TABLE ... AS SELECT` statements.
  - `CREATE MATERIALIZED VIEW ... AS SELECT` statements.

- Each time an `UPDATE` statement updates one row, the sequence is incremented once.
- For each `INSERT` statement that contains a `VALUES` clause, the sequence is incremented once.
- Each time a `MERGE` statement merges one row, the sequence is incremented once. `NEXTVAL` can appear in the `merge_insert_clause` or `merge_update_clause` clause or both. `NEXTVAL` is incremented as each row is updated and inserted, even if the sequence values are not used for the update or insert operation. If `NEXTVAL` is specified multiple times in these positions, the sequence is incremented once for each row. The same value is returned for all the occurrences of `NEXTVAL` in the row.

If the `NEXTVAL` pseudocolumn of a sequence is referenced multiple times in these positions, the sequence is incremented only once. That is, the next sequence value of the current sequence is returned for all the `NEXTVAL` pseudocolumns that are referenced.

If both `CURRVAL` and `NEXTVAL` pseudocolumns of a sequence are referenced in these positions, ApsaraDB for OceanBase increments the sequence. That is, the next sequence value of the current sequence is returned for the `CURRVAL` and `NEXTVAL` pseudocolumns that are referenced.

Multiple users can access a sequence at the same time without waiting and locking.

ROWSCN pseudocolumn

The `ORA_ROWSCN` pseudocolumn reflects the latest System Change Number (SCN) to a row. The SCN indicates the commit time of the transaction that modifies the data of this row.

ROWNUM pseudocolumn

The `ROWNUM` pseudocolumn numbers each row in the query result. The value indicates the specific position of the row in the query result set. The value 1 is returned for the first row, 2 for the second row, and so on.

Instructions on the usage of ROWNUM

`ROWNUM` can limit the number of returned rows. In the following example, five data records in the **employees** table are returned.

```
SELECT * FROM employees WHERE rownum <=5;
```

If an `ORDER BY` clause follows `ROWNUM`, the results that meet the `WHERE` conditions are resorted. Assume that you embed an `ORDER BY` clause in a subquery and use the `ROWNUM` pseudocolumn as a condition in the top-level query. In this case, you can force the `ROWNUM` condition to be applied after the rows are sorted. For example, you cannot obtain the expected result if you execute the following statement to query information about the five oldest employees. The statement only sorts the first five employee information entries in the query result by age:

```
SELECT * FROM employees WHERE rownum <=5 ORDER BY age DESC;
```

The following statement is valid:

```
SELECT * FROM (SELECT * FROM employees ORDER BY age DESC) WHERE rownum <= 5;
```

If you specify that `ROWNUM` is greater than a positive integer in the `WHERE` clause, **FALSE** is always returned. For example, the following SQL statement returns no information:

```
SELECT * FROM employees WHERE rownum > 1;
```

This is because when the result of the first row in the table is retrieved, the value of the `ROWNUM` pseudocolumn of this row is assigned 1. In this case, the result is **FALSE** when the `WHERE` condition is checked. This row is discarded. When the result of the second row is retrieved, the value of the `ROWNUM` pseudocolumn of this row is still assigned 1. The result is still **FALSE** when the `WHERE` condition is checked. This row is also discarded. This way, all the rows fail to meet the condition. Therefore, no data is returned.

You can also assign the `ROWNUM` values to a column in the table by executing a `UPDATE` statement, as shown in the following example.

```
UPDATE employees SET id = rownum;
```

The statement assigns the `ROWNUM` values to the **id** column in the **employees** table. The **id** column is assigned values 1, 2,... until the total number of rows in the table is reached.

> **Notice**
>
> If you use `ROWNUM` in queries, view optimization may be affected.

17.1.5.4. Elements

17.1.5.4.1. Built-in data types

17.1.5.4.1.1. Overview of built-in data types

Each value manipulated by ApsaraDB for OceanBase has a data type. The value's data type associates a fixed set of attributes with the value. These attributes cause ApsaraDB for OceanBase to treat values of one data type differently from values of another. ApsaraDB for OceanBase provides a number of built-in data types. They are also called basic data types of ApsaraDB for OceanBase.

Data types

ApsaraDB for OceanBase supports the following data types, which are consistent with the Oracle data types:

- Character data types
- Numeric data types
- Date and time data types
- RAW data types
- Large object data types

17.1.5.4.1.2. Character data types

Overview of character data types

Character data types store character (alphanumeric) data, which are words and free-form text, in the database character set or national character set. Character data types are subject to more limits than other data types. Therefore, character data types have fewer attributes.

Character data is stored in strings with byte values that correspond to one of the character sets. The character sets are specified when a database is created. ApsaraDB for OceanBase supports both single-byte and multibyte character sets.

> **Note**
>
> The columns of character data types can store all alphanumeric values. But the columns of `NUMBER` data types can store only numeric values.

Data type	Length	Description	Length description
CHAR(size [BYTE | CHAR])	Fixed length	High index efficiency. The program uses the `trim` function to remove white spaces.	Set the `size` parameter to a numeric value between 1 and 2000. The number of bytes for storage is the value of the `size` parameter.
NCHAR[(size)]	Fixed length	Use the Unicode character set. Two bytes are required to represent all characters.	Set the `size` parameter to a numeric value between 1 and 2000. The number of bytes for storage is twice the value of the `size` parameter.
NVARCHAR2(size)	Variable length	Use the Unicode character set. Two bytes are required to represent all characters.	Set the `size` parameter to a value between 1 and 32767. The number of bytes for storage is twice that of input characters.
VARCHAR2(size [BYTE | CHAR])	Variable length	Use the Unicode character set. Two bytes are required to represent all characters.	Set the `size` parameter to a numeric value between 1 and 32767. The number of bytes for storage is that of bytes for input characters rather than the value of the `size` parameter.

Data type	Length	Description	Length description
VARCHAR(size [BYTE | CHAR])	Variable length	In ApsaraDB for OceanBase, `VARCHAR` functions the same as `VARCHAR2`.	Set the `size` parameter to a numeric value between 1 and 32767. The number of bytes for storage is that of bytes for input characters rather than the value of the `size` parameter.

You must specify the `length` semantics for the `CHAR` and `VARCHAR2` data types. The default length semantics is defined by the `NLS_LENGTH_SEMANTICS` system variable.

CHAR data type

The `CHAR` data type stores fixed-length strings. ApsaraDB for OceanBase ensures that all values stored in a `CHAR` column have the fixed length specified by `size`. If you insert a value that is shorter than the specified length, ApsaraDB for OceanBase pads the value to the specified length with spaces. If you insert a value that exceeds the specified length, ApsaraDB for OceanBase returns an error.

The `BYTE` and `CHAR` qualifiers override the semantics specified by the `NLS_LENGTH_SEMANTICS` parameter, which defaults to the BYTE semantics. To ensure proper data conversion between databases that use different character sets, the `CHAR` data must consist of well-formed strings.

Syntax

```
CHAR [(size [BYTE | CHAR])]
```

Parameters

Parameter	Description
size	A fixed length.
BYTE	Specifies that the column length is measured in bytes.
CHAR	Specifies that the column length is measured in characters.

The default length of a `CHAR` column is 1 byte. The maximum length is 2,000 bytes.

Examples

By default, when you create a table that contains a `CHAR` column, you specify the column length in bytes. `BYTE` is the default qualifier.

If you use the `CHAR` qualifier, you specify the column length in characters.

- **Example 1**

Assume that you insert a 1-byte string into the `CHAR(10)` column. Before the string is stored, the system pads the string to 10 bytes with spaces.

- **Example 2**

`CHAR(10 CHAR)` indicates that you specify the column length in characters.

NCHAR data type

The `NCHAR` data type specifies the fixed-length `UNICODE` character data. When you create a database, the national character set defines the maximum column length. When you create a table that contains an `NCHAR` column, you define the column length in characters. Width specifications of the character data type `NCHAR` indicate the number of characters. The maximum column length is 2,000 bytes.

If you want to use less space to store Chinese characters, choose the `NCHAR` data type.

When you use an `NCHAR` column to store values, the database automatically pads the values that are shorter than the specified length with spaces to the specified length. When you specify lengths, `CHAR` is used as the unit of measurement. You cannot specify other units.

> **Notice**
>
> You cannot insert a `CHAR` value into an `NCHAR` column or insert an `NCHAR` value into a `CHAR` column.

Syntax

```
NCHAR[(size)]
```

Parameters

Parameter	Description
`size`	The length of a fixed-length character string. Based on the national character set, the maximum length is set to 2,000 bytes. By default, the minimum length of a fixed-length character string is one character.

More information

Unicode character set

The Unicode character set is an encoding of characters. It provides UTF-8, UTF-16, UTF-32, and other compression and conversion encoding methods. An encoding method determines the size required to store a character. Chinese characters and English characters take up different spaces varying from storage methods.

Comparison of three encoding methods

Encoding method	Number of bytes for encoding characters	BOM	Advantage	Disadvantage
UTF-8	A variable-length encoding method that provides single-byte encoding for ASCII characters and multibyte encoding for non-ASCII characters. The minimum code unit is eight bits.	Without BOM: If a byte stream starts with EF BB BF at the beginning of a text, the text is encoded in UTF-8.	An ideal Unicode encoding method: This method is fully compatible with ASCII encoding, requires no BOM, features strong self-synchronization and error correction capabilities for network transmission and communication, and provides high scalability.	The variable-length encoding makes internal processing of the program more difficult.
UTF-16	Two or four bytes. The minimum code unit is 16 bits.	With BOM: UTF-16LE (little-endian) is represented by FF FE, and UTF-16BE (big-endian) is represented by FE FF.	The earliest Unicode encoding method that has been applied to various scenarios. This method is suitable for Unicode processing in memory, and is used to encode strings in APIs across multiple programming languages.	This method is not compatible with ASCII encoding, and has poor scalability. The encoding is complex when surrogate pairs are used to encode code points in the supplementary planes.
UTF-32	A fixed length of four bytes. The minimum code unit is 16 bits.	With BOM: UTF-16LE (little-endian) is represented by FF FE, and UTF-16BE (big-endian) is represented by FE FF.	A fixed-byte encoding that is easy to read and is internally processed by a compiler. This method provides a one-to-one mapping between Unicode code points and code units.	All characters are encoded in a fixed length of four bytes, which wastes storage space and bandwidth. This method is not compatible with ASCII encoding, has a poor scalability, and is not used in most cases.

Database character set

- Used to store the data types such as `CHAR`, `VARCHAR2`, and `CLOB`
- Used to identify the information such as table names, column names, and PL/SQL variables
- Used to store SQL and PL/SQL program units

National character set

- Used to store the data types such as `NCHAR`, `NVARCHAR2`, and `NCLOB`
- The national character set is essentially an additional character set that is selected for ApsaraDB for OceanBase. The national character set is mainly used to enhance the character processing capability of ApsaraDB for OceanBase. The `NCHAR` data type uses the national character set. While using the database character set provided by the `CHAR` data type, the `NCHAR` data type provides an alternative to the database character set.

NVARCHAR2 data type

The `NVARCHAR2` data type stores `UNICODE` characters. An NVARCHAR2 column stores variable-length values. The maximum length of an NVARCHAR2 column is 32,767 bytes. The minimum length of an NVARCHAR2 column is 1 byte. When you create a table that contains an `NVARCHAR2` column, you must specify the maximum number of characters that a value in the `NVARCHAR2` column can contain. By default, `CHAR` is the unit that is used to measure the column length. You cannot specify other units.

If the values to be stored have uncertain lengths, you can choose the `NVARCHAR2` type.

Format

```
NVARCHAR2(size)
```

Parameters

Parameter	Description
size	A variable column length. You must specify the size of `NVARCHAR2`. The number of bytes can be up to two times the specified size if AL16UTF16 encoding is used and three times the specified size if UTF8 encoding is used. The national character set defines the number of bytes. The upper limit is 32,767 bytes.

VARCHAR2 data type

The `VARCHAR2` data type stores variable-length strings. The maximum string length is 32,767 bytes. When you create a `VARCHAR2` column, you must specify the maximum length for the `VARCHAR2` column. Although you can store a zero-length string (''), the maximum value must be at least 1 byte. ApsaraDB for OceanBase stores each value in the column in the way as you specify.

Notice

- From a technical point of view, a character is a code point of the database character set.
- Assume that you create a database object with a VARCHAR2 column or attribute. If you specify no explicit qualifiers in the column or attribute definition, the value of the `NLS_LENGTH_SEMANTICS` parameter of the session determines the length semantics.
- ApsaraDB for OceanBase compares the `VARCHAR2` values by using non-padded comparison semantics.
- When you convert data between databases that use different character sets, you must ensure that the `VARCHAR2` data consists of well-formed strings.

Syntax

```
VARCHAR2(size [BYTE | CHAR])
```

Parameters

Parameter	Description
size	The number of the stored bytes or characters.
BYTE	Specifies that the column length is measured in bytes.
CHAR	Specifies that the column length is measured in characters.

Examples

You can specify the maximum length in characters by using the `CHAR` qualifier.

```
VARCHAR2(10 CHAR)
```

You can specify the maximum length in bytes by using the `BYTE` qualifier.

```
VARCHAR2(10 BYTE)
```

VARCHAR data type

The `VARCHAR` data type is used to store variable-length strings. When you create a `VARCHAR` column, you must specify the maximum length for the `VARCHAR` column. Although you can store a zero-length string (''), the maximum length must be at least 1 byte. ApsaraDB for OceanBase stores each value in the column in the way as you specify. If you insert a value that exceeds the specified length, ApsaraDB for OceanBase returns an error.

In ApsaraDB for OceanBase, the `VARCHAR` data type is synonymous with the `VARCHAR2` data type. In most cases, `VARCHAR2` is used.

Syntax

```
VARCHAR(size [BYTE | CHAR])
```

Parameters

Parameter	Description
size	The number of the stored bytes or characters.
BYTE	Specifies that the column length is measured in bytes.
CHAR	Specifies that the column length is measured in characters.

17.1.5.4.1.3. Numeric data types

Overview of numeric data types

ApsaraDB for OceanBase allows you to store numeric values in the `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE` data types. You can store fixed-point numbers, floating-point numbers, and zero by using these numeric data types. In numeric calculations, the numeric data types have different operator precedence values. For more information, see Precedence of numeric data types.

Data type	Length (bytes)	Description
NUMBER	4~40	`NUMBER(p,s)` stores variable-length fixed-point numbers that have decimal precisions. You can also store floating-point numbers by using `NUMBER` where `p` and `s` are not specified.
FLOAT	4~40	`FLOAT(p)` is a subtype of the `NUMBER` data type. The binary precision ranges from 1 to 126. A value of `FLOAT` is not a floating-point number.
BINARY_FLOAT	4	This data type stores 32-bit single-precision floating-point numbers that have binary precisions.
BINARY_DOUBLE	8	This data type stores 64-bit double-precision floating-point numbers that have binary precisions.

NUMBER data type

`NUMBER` is a numeric data type that stores variable-length and exact values. Each value of this data type occupies 4 to 40 bytes of storage space. The storage space of 4 bytes is used to store `NUMBER` metadata and the storage space of 36 bytes is used to store specific `NUMBER` values. The NUMBER data type can store zeros, floating-point numbers, positive fixed-point numbers, and negative fixed-point numbers. The absolute values that can be stored range from 1.0×10^{-130} to 1.0×10^{126} (excluding 1.0×10^{126}). ApsaraDB for OceanBase returns an error if the absolute value of the specified arithmetic expression is greater than or equal to 1.0×10^{-130}.

The `NUMBER` data type provides higher data accuracy and is more general and portable than the floating-point data type. The floating-point data type enables more efficient arithmetic operations than the NUMBER data type.

Syntax

```
NUMBER [(p[s])]
```

Parameters

Parameter	Value range	Description
p	1~38	Specifies the precision. The value of this parameter is the maximum number of significant decimal digits. The most significant digit is the leftmost non-zero digit and the least significant digit is the rightmost known digit.
s	-84~127	Specifies the number of digits in the fractional part. The value of this parameter is the number of digits from the decimal point to the least significant digit. The scale range is from -84 to 127.

Note

- If s is greater than 0, the value is rounded to `s` digits on the right of the decimal point. Then, the system checks whether the number of significant digits is smaller than or equal to `p`.
- If s is smaller than 0, the value is rounded to `s` digits on the left of the decimal point. Then, the system checks whether the number of significant digits is smaller than or equal to `p + |s|`.
- If s is equal to 0, the value is an integer.

> **Notice**
>
> - The positive scale for decimal places is the number of significant digits from the first digit on the right of the decimal point to the least significant digit. Precisions and decimal places are represented by decimal digits.
> - The negative scale for decimal places is the number of significant digits on the left of the decimal point, excluding the least significant digit. For the negative scale, the least significant digit is on the left of the decimal point because the actual data is rounded to the specified number of digits on the left of the decimal point.

Examples

- **Example 1:** Use the following format to specify an integer:

`NUMBER(p)` specifies a fixed-point number with a precision of `p` and a scale of 0. NUMBER(p) is equivalent to `NUMBER(p,0)`.

`NUMBER` specifies a floating-point number. In this format, the designators for precisions and decimal places are absent.

- **Example 2:** Store data that has different precisions and different numbers of decimal places. To prevent the precisions of the data stored in ApsaraDB for OceanBase from exceeding the specified precisions, specify the precisions and the number of decimal places for fixed-point number columns. You must also perform integrity checks on the input data. However, this does not enforce a fixed length of the values in the fixed-point number columns. If the precisions of the actual data to be stored exceed the specified precisions, ApsaraDB for OceanBase returns an error. If the number of decimal places of the actual data to be stored exceeds the specified value, ApsaraDB for OceanBase rounds the data.

Actual data	Specified As	Stored As
123.89	`NUMBER`	123.89
123.89	`NUMBER(3)`	124
123.89	`NUMBER(3,2)`	Exceeds the specified precision
123.89	`NUMBER(4,2)`	Exceeds the specified precision
123.89	`NUMBER(5,2)`	123.89
123.89	`NUMBER(6,1)`	123.9
123.89	`NUMBER(6,-2)`	100
.01234	`NUMBER(4,5)`	.01234
.00012	`NUMBER(4,5)`	.00012

Actual data	Specified As	Stored As
.000127	`NUMBER(4,5)`	.00013
.000012	`NUMBER(2,7)`	.000012
.0000123	`NUMBER(2,7)`	.000012
1.2e-4	`NUMBER(2,5)`	.00012
1.2e-5	`NUMBER(2,5)`	.00001

FLOAT data type

The `FLOAT` data type is a subtype of the `NUMBER` data type that is specified with precision. This data type occupies 4 bytes to 40 bytes of storage space. The precision is calculated based on the number of significant binary digits which ranges from 1 to 126. The number of decimal places is not user-defined. The `FLOAT` data type specifies the variable-length non-exact numeric data.

Syntax

```
FLOAT [(p)]
```

Parameters

Parameter	Definition	Value range	Description
p	Precision	1~126	The precision for numeric values. The precision is calculated based on the number of significant binary digits. To convert from binary precision to decimal precision, you must multiply the binary precision by 0.30103.

Note

- Binary-to-decimal precision conversion: `Binary precision = int(Decimal precision x 0.30103)`
- Decimal-to-binary precision conversion: `Decimal precision = int(Binary precision x 3.32193)`

Examples

- **Example 1**: Assume that you set the binary precision to 2 by using `FLOAT`, convert the binary precision to the decimal precision by using the following function: int(2 x 0.30103) = 0.6, and round down the result to an integer. The decimal precision of `FLOAT(2)` is 0.

```
FLOAT(2)
```

- **Example 2**: Assume that you create a table named **test**, and insert data into the table. The **col1** column is of the `NUMBER` data type, and the **col2** column is of the `FLOAT` data type. `NUMBER(5,2)` represents a fixed-point value with decimal precision. The result is limited to five significant digits, and is returned with two decimal places. `FLOAT(5)` represents the binary precision of 5. This value is converted to the decimal precision by using the following function: `int(5 x 0.30103) = 1.50515`. After the result is rounded down to an integer, the decimal precision is 1. For example, 123.45 is expressed as 1.2345×10^2 in scientific notation. If one decimal place is required, the value of 1.2345 is rounded to 1.2. The following result is finally displayed: $1.2 \times 10^2 = 120$. Execute the following statement:

```
CREATE TABLE test (col1 NUMBER(5,2), col2 FLOAT(5));
INSERT INTO test VALUES (1.23, 1.23);
INSERT INTO test VALUES (7.89, 7.89);
INSERT INTO test VALUES (12.79, 12.79);
INSERT INTO test VALUES (123.45, 123.45);
```

Execute the following statement to query the **test** table:

```
SELECT * FROM test;
```

The following result is returned:

```
+-----------+----------+
| col1          |   col2   |
+-----------+----------+
|    1.23   |   1.2    |
+-----------+----------+
|    7.89   |   7.9    |
+-----------+----------+
|    12.79  |   13     |
+-----------+----------+
|    123.45 |   120    |
+-----------+----------+
```

> **Note**
>
> When you convert the `ANSI FLOAT` data, you can use the `FLOAT` data type that is used by ApsaraDB for OceanBase. However, we recommend that you use `BINARY_FLOAT` and `BINARY_DOUBLE` for floating-point numbers in ApsaraDB for OceanBase.

Floating-point numbers

Floating-point numbers can have a decimal point anywhere from the first to the last digit, or can have no decimal point at all. You can choose to optionally use an exponent following the number to increase the range, for example, 1.666 e^{-20}. Decimal places are not applicable to floating-point numbers, because the number of digits that can appear after the decimal point is not limited.

> **Notice**
>
> Binary floating-point numbers differ from `NUMBER` in the way where the values are internally stored by ApsaraDB for OceanBase. All values of the `NUMBER` data type are exactly stored with decimal precision. Binary floating-point numbers are stored with binary precision by using the digits 0 and 1. Such a storage method cannot exactly represent all values that use decimal precision.

Syntax

ApsaraDB for OceanBase provides two numeric data types for floating-point numbers:

- `BINARY_FLOAT` is a 32-bit single-precision floating-point number data type. Each `BINARY_FLOAT` value requires 4 bytes.
- `BINARY_DOUBLE` is a 64-bit double-precision floating-point number data type. Each `BINARY_DOUBLE` value requires 8 bytes.

> **Note**
>
> - In a `NUMBER` column, floating-point numbers have decimal precision.
> - In a `BINARY_FLOAT` or `BINARY_DOUBLE` column, floating-point numbers have binary precision.
> - The binary floating-point numbers do not support the special values `infinity` and `NaN` (not a number).

Value range

You can specify floating-point numbers within the value range.

Value	BINARY_FLOAT	BINARY_DOUBLE
Maximum positive finite value	3.40282E+38F	1.79769313486231E+308
Minimum positive finite value	1.17549E-38F	2.22507485850720E-308

More information

IEEE754 conformance

- IEEE Standard 754-1985 (IEEE754)

The implementation of floating-point data types in ApsaraDB for OceanBase conforms substantially with the Institute of Electrical and Electronics Engineers (IEEE) Standard for Binary Floating-Point Arithmetic.

- The floating-point data types conform to `IEEE754` in the following areas:
  - The SQL function `SQRT` returns the square root.

- The SQL function `REMAINDER` returns the remainder.
- Arithmetic operators conform to the standard.
- Comparison operators conform to the standard.
- Conversion operators conform to the standard.
- The default rounding mode is supported.
- The default exception handling mode is supported.
- ApsaraDB for OceanBase does not support the special constants such as `INF`, `-INF`, `NaN`, `BINARY_FLOAT_NAN`, and `BINARY_DOUBLE_NAN`.
- The SQL functions `ROUND`, `TRUNC`, `CEIL`, and `FLOOR` enable the rounding of binary floating-point numbers of `BINARY_FLOAT` and `BINARY_DOUBLE` to integer values of `BINARY_FLOAT` and `BINARY_DOUBLE`.
- The SQL functions `TO_CHAR`, `TO_NUMBER`, `TO_NCHAR`, `TO_BINARY_FLOAT`, `TO_BINARY_DOUBLE`, and `CAST` enable the rounding of binary floating-point numbers of `BINARY_FLOAT` and `BINARY_DOUBLE` to decimal values, and the rounding of decimal values to binary floating-point numbers of `BINARY_FLOAT` and `BINARY_DOUBLE`.

- The floating-point data types do not conform to IEEE754 in the following areas:
  - The value of -0 is coerced to that of +0.
  - Comparison with `NaN` is not supported.
  - Non-default rounding modes are not supported.
  - Non-default exception handling modes are not supported.

Numeric precedence

Different numeric data types have different levels of precedence in operations. In ApsaraDB for OceanBase, `BINARY_DOUBLE` has the highest precedence, followed by `BINARY_FLOAT`, and finally by `NUMBER`.

In any operation on multiple numeric values:

- If an operand is `BINARY_DOUBLE`, ApsaraDB for OceanBase converts all operands to `BINARY_DOUBLE` before performing the operation.
- If an operand is `BINARY_FLOAT`, ApsaraDB for OceanBase converts all operands to `BINARY_FLOAT` before performing the operation.
- If none of the operands is `BINARY_DOUBLE` or `BINARY_FLOAT`, ApsaraDB for OceanBase converts all operands to `NUMBER` before performing the operation.
- If the required conversion fails, the operation fails.
- In the context of other data types, numeric data types have lower precedence than the date, time, and interval data types, and have higher precedence than character and all other data types.

17.1.5.4.1.4. Date, time, and interval data types

Overview of datetime and interval data types

ApsaraDB for OceanBase supports datetime and interval data types. These data types are the same as those in Oracle.

Datetime data types

The datetime data types are used to store date and time data that is stored in databases. This category consists of the following data types.

Data type	Description
DATE	Stores date and time information. The precision is seconds. The information about time zones is excluded.
TIMESTAMP	Serves as an extension of the `DATE` data type. The precision is nanoseconds. The information about time zones is excluded.
TIMESTAMP WITH TIME ZONE	Serves as an extension of the `DATE` data type. The precision is nanoseconds. The information about time zones is included.
TIMESTAMP WITH LOCAL TIME ZONE	Stores the information about database time zones. This data type stores `TIMESTAMP` values that contain the information about the local time zone.

Interval data types

Interval data types are different from datetime data types in the following aspect: Datetime data types store specific time points and interval data types store time periods. Interval data types provide an effective way to store the difference between each two datetime values. This category consists of the following data types.

Data type	Description
INTERVAL YEAR TO MONTH	Stores time periods that are measured in years and months.
INTERVAL DAY TO SECOND	Stores time periods that are measured in days, hours, minutes, and seconds.

More information

- Daylight saving time
- Calculation of DATE and INTERVAL values

DATE data type

The `DATE` data type stores the date and time information. Although you can use both character and numeric data types to represent the date and time information, the `DATE` data type has special associated properties.timetime For each `DATE` value, ApsaraDB for OceanBase stores the information about year, month, day, hour, minute, and second, but does not contain the time zone information.time zoneinfo

Format

`NLS_DATE_FORMAT` determines the default input and output formats of the `DATE` data type. Run the following SQL statement to query the default format:

```
SELECT @@NLS_DATE_FORMAT FROM DUAL;
```

Return result:

```
DD-MON-RR
```

To customize data formats, you can use conversion functions. When you insert data, you can use the `TO_DATE(char,fmt)` function to specify the input format of the data. When you query data, you can use the `TO_CHAR(datetime,fmt)` function to specify the output format of the data. These two conversion functions convert a string to the format defined by the `fmt` parameter. If you do not specify `fmt`, the default format is used.

> **Notice**
>
> The `DATE` data type stores the time information such as the hour, minute, and second. However, the default format does not contain the time information.

Value range

`0001-01-01 00:00:00` ~ `9999-12-31 23:59:59`

Examples

- **Example 1**: This example returns the current system date. Assume that you do not specify the `fmt` parameter, the `TO_CHAR` function returns a value in the default format of the data type.

```
SELECT TO_CHAR(sysdate) FROM DUAL;
```

Return result:

```
+------------------+
| TO_CHAR(SYSDATE) |
+------------------+
| 24-FEB-20        |
+------------------+
```

- **Example 2**: Assume that you do not specify the `DATE` value as a literal, the database returns the system default value:

- Year: indicates the current year, which is returned by `SYSDATE`.
- Month: indicates the current month, which is returned by `SYSDATE`.
- Day: defaults to 01 that indicates the first day of the current month.
- The hour, minute, and second values are all 0.

This example uses the `TO_DATE(string,format)` function to insert date data. `TO_DATE` converts the characters in the `string` parameter to a value in the format specified in the `format` parameter.

Assume that you issue the following query in February 2020:

```
SELECT TO_CHAR(TO_DATE('2020', 'YYYY'),'YYYY-MM-DD HH24:MI:SS') FROM DUAL;
```

Return result:

```
+-----------------------------------------------------------+
| TO_CHAR(TO_DATE('2020', 'YYYY'),'YYYY-MM-DD HH24:MI:SS') |
+-----------------------------------------------------------+
|                                      2020-02-01 00:00:00|
+-----------------------------------------------------------+
```

TIMESTAMP data type

In addition to the `DATE` data type, the date and time data types also include the `TIMESTAMP[(scale)]` data type. The TIMESTAMP data type is an extension of the `DATE` data type. The TIMESTAMP data type stores some information that the `DATE` data type stores, such as the year, month, day, hour, minute, and second values. However, the TIMESTAMP data type does not store the time zone information. The highest precision of a TIMESTAMP value is nanosecond precision. Therefore, this data type is often used to store the data that has a high time precision and does not require time zone conversion.

Syntax

```
TIMESTAMP [(scale)]
```

Parameters

Parameter	Value range	Description
scale	0~9	The value of `scale` determines the range and precision of `TIMESTAMP[(scale)]`. The maximum value of scale is 9, which means nanosecond precision. If scale is 9, nine digits appear after the decimal point of the second value. The minimum value of scale is 0, which indicates second precision. If scale is 0, no digits appear after the decimal point of the second value. The default value of scale is 6.

Format

`NLS_TIMESTAMP_FORMAT` determines the default input and output formats of the `TIMESTAMP` data type. Execute the following SQL statement to query the default format:

```
SELECT @@NLS_TIMESTAMP_FORMAT FROM DUAL;
```

The following result is returned:

```
DD-MON-RR HH.MI.SSXFF AM
```

To customize data formats, use conversion functions. When you insert data, you can use the `TO_TIMESTAMP(char,fmt)` function to specify the input format of the data. When you query data, you can use the `TO_CHAR(datetime,fmt)` function to specify the output format of the data. These two conversion functions convert a string to the format defined by the `fmt` parameter. If you do not specify `fmt`, the default format is used.

Value range

`0001-01-01 00:00:00.000000000` ~ `9999-12-31 23:59:59.999999999`

Examples

- **Example 1**: You can run the following code to create two **TIMESTAMP** columns named **timestp1** and **timestp2** in the `Timestamp_Sample` table and set scale to 3 for the **timestp2** column. The date value **2020-01-01 11:00:00** is inserted into both columns in the format of `TO_TIMESTAMP(string,format)`.

```
CREATE TABLE Timestamp_Sample(timestp1 TIMESTAMP, timestp2 TIMESTAMP(3));
INSERT INTO Timestamp_Sample(timestp1,timestp2) VALUES(TO_TIMESTAMP('2020-01-01 11:00:00','YYYY-MM-DD
HH24:MI:SS'),TO_TIMESTAMP('2020-01-01 11:00:00','YYYY-MM-DD HH24:MI:SS'));
SELECT * FROM Timestamp_Sample;
```

The following result is returned. The **timestp1** column uses the default time precision because the value of `scale` is not specified. Six digits appear after the decimal point of the second value. However, in the **timestp2** column, three digits appear after the decimal point of the second value because scale is set to 3.

```
+-----------------------------+----------------------------+
| timestamp1                  | timestamp2                 |
+-----------------------------+----------------------------+
| 01-JAN-20 11.00.00.000000 AM | 01-JAN-20 11.00.00.000 AM |
+-----------------------------+----------------------------+
```

- **Example 2**: In the following statement, the `TO_CHAR(datetime,fmt)` function is used to specify the output format:

```
SELECT TO_CHAR(TO_TIMESTAMP_TZ('25-FEB-20 11:00:00 AM America/Los_Angeles','DD-MON-RR HH:MI:SSXFF PM
TZR'),'YYYY-MM-DD HH:MI:SSXFF PM TZR') Timestamp
FROM DUAL;
```

The following result is returned:

```
+-----------------------------------------------------+
| Timestamp                                           |
+-----------------------------------------------------+
| 2020-02-25 11:00:00.000000000 AM America/Los_Angeles |
+-----------------------------------------------------+
```

TIMESTAMP WITH TIME ZONE data type

`TIMESTAMP [(scale)] WITH TIME ZONE` is a variant of `TIMASTAMP [(scale)]`. TIMESTAMP [(scale)] WITH TIME ZONE stores all the time information that is stored in a `TIMASTAMP [(scale)]` value, such as the year, month, day, hour, minute, and second values. The value of `scale` determines the bounds and precision of TIMESTAMP [(scale)] WITH TIME ZONE. The TIMESTAMP [(scale)] WITH TIME ZONE data type also stores the time zone information that `TIMASTAMP [(scale)]` does not store. Therefore, TIMESTAMP [(scale)] WITH TIME ZONE is often used to store the date and time information across geographical regions.

Syntax

```
TIMESTAMP [(scale)] WITH TIME ZONE
```

Parameters

Parameter	Value range	Description
scale	0~9	The value of `scale` determines the bounds and precision of `TIMESTAMP [(scale)] WITH TIME ZONE`. The maximum value of scale is 9, which indicates nanosecond precision. If scale is 9, nine digits appear after the decimal point of the second value. The minimum value of scale is 0, which indicates second precision. If scale is 0, no digits appear after the decimal point of the second value. The default value of scale is 6.

Format

`NLS_TIMESTAMP_TZ_FORMAT` determines the default input and output formats of the `TIMESTAMP WITH TIME ZONE` data type. Execute the following SQL statement to query the default formats:

```
SELECT @@NLS_TIMESTAMP_TZ_FORMAT FROM DUAL;
```

The following result is returned:

```
DD-MON-RR HH.MI.SSXFF AM TZR
```

To customize data formats, use conversion functions. When you insert data, you can use the `TO_TIMESTAMP_TZ(char,fmt)` function to specify the input format of the data. When you query data, you can use the `TO_CHAR(datetime,fmt)` function to specify the output format of the data. These two conversion functions convert a string to the format defined by the `fmt` parameter. If you do not specify `fmt`, the default format is used.

Value range

`0001-01-01 00:00:00.000000000` ~ `9999-12-31 23:59:59.999999999`

Examples

In the following examples, `TO_TIMESTAMP_TZ(char,fmt)` is used to insert a timestamp value.

When you insert a time zone, ApsaraDB for OceanBase allows you to use the offset and region name of the time zone.

- The time zone offset is the difference (in hours and minutes) between the local time zone and the UTC+0 time zone.
- Specify the time zone region name (TZR) and time zone abbreviation (TZD) in this format: Country/City Time zone abbreviation.

Use the time zone offset

Execute the following statement to insert a value by using the time zone offset:

```
SELECT TO_TIMESTAMP_TZ('2020-01-01 11:00:00 -05:00','YYYY-MM-DD HH:MI:SS TZH:TZM') FROM DUAL;
```

The following result is returned:

```
01-JUN-20 11.00.00.000000000 AM AMERICA/LOS_ANGELES
```

Use the region name and abbreviation of the time zone

Execute the following statement to insert a value by using the region name and abbreviation of the time zone:

```
SELECT TO_TIMESTAMP_TZ('2020-01-01 11:00:00 America/Los_Angeles PST','YYYY-MM-DD HH:MI:SS TZR TZD') F
ROM DUAL;
```

The following result is returned:

```
01-JUN-20 11.00.00.000000000 AM America/Los_Angeles PST
```

Daylight saving time

ApsaraDB for OceanBase supports daylight saving time and uses time zone abbreviations to indicate the information about daylight saving time. For example, in the time zone region America/Los_Angeles, the daylight saving time is Pacific Daylight Time (PDT). PDT is in effect from the second Sunday in March to the first Sunday in November each year. Pacific Standard Time (PST) is in effect in the other periods of each year. If an inserted value contains only the name of the time zone region, ApsaraDB for OceanBase determines whether PDT applies to the inserted time zone region based on the inserted time information. The output includes the time zone abbreviation to indicate whether the current time is in PDT.

You can run the following sample code:

```
SELECT TO_TIMESTAMP_TZ('2020-02-01 11:00:00 America/Los_Angeles','YYYY-MM-DD HH:MI:SS TZR') FROM DUAL
;
SELECT TO_TIMESTAMP_TZ('2020-06-01 11:00:00 America/Los_Angeles','YYYY-MM-DD HH:MI:SS TZR') FROM DUAL
;
```

The following result is returned:

```
01-JUN-20 11.00.00.000000000 AM America/Los_Angeles PST
01-JUN-20 11.00.00.000000000 AM America/Los_Angeles PDT
```

TIMESTAMP WITH LOCAL TIME ZONE data type

The time zone in a value of the `TIMESTAMP [(scale)] WITH LOCAL TIME ZONE` data type is the time zone where the current session occurs. When you specify a `TIMASTAMP [(scale)] WITH TIME ZONE` value, you must specify the time zone. However, when you specify a TIMESTAMP [(scale)] WITH LOCAL TIME ZONE value, ApsaraDB for OceanBase automatically stores the default database time zone (UTC+0). You cannot change the time zone. When you retrieve data, ApsaraDB for OceanBase returns the time zone of the current session and you can change the time zone. This data type is often used to store the date information that is always displayed in the time zone of the client system in a two-tier application.

Syntax

```
TIMASTAMP [(scale)] WITH TIME ZONE
```

Parameters

Parameter	Value range	Description
scale	0~9	The value of `scale` determines the bounds and precision of `TIMESTAMP [(scale)]`. The maximum value is 9, which indicates nanosecond precision. The minimum value is 0, which indicates second precision. The default value is 6.

Format

`NLS_TIMESTAMP_FORMAT` determines the default input and output formats of the `TIMESTAMP WITH LOCAL TIME ZONE` data type. Execute the following SQL statement to query the date and time format:

```
SELECT @@NLS_TIMESTAMP_FORMAT FROM DUAL;
```

The following result is returned:

```
DD-MON-RR HH.MI.SSXFF AM
```

To customize data formats, use conversion functions. You can use the `TO_CHAR(datetime,fmt)` function to specify the output data format. The conversion function converts a string to the format defined by the fmt parameter. If you do not specify `fmt`, `TO_CHAR` returns data in the default format of the data type. For more information about the input formats of the `TIMESTAMP WITH TIME ZONE` data type, see Timestamp literals.

Value range

`0001-01-01 00:00:00.000000000` ~ `9999-12-31 23:59:59.999999999`

Examples

No literals can be used to specify a `TIMESTAMP WITH TIME ZONE` value. **SESSIONTIMEZONE** returns the information of the local session time zone. The value of **SESSIONTIMEZONE** is the same as the value of the custom parameter **TIME_ZONE**.

```
CREATE TABLE LocalTZ ( ltzcol TIMESTAMP WITH LOCAL TIME ZONE);
INSERT INTO LocalTZ VALUES (TIMESTAMP '2020-02-25 11:10:08.123');
ALTER SESSION SET TIME_ZONE='+08:00';
SELECT SESSIONTIMEZONE, ltzcol FROM LocalTZ;
```

The following result is returned:

```
+-----------------+-------------------------------+
| SESSIONTIMEZONE | ltzcol                        |
+-----------------+-------------------------------+
| +08:00          | 25-FEB-20 11:10:08.123000 AM |
+-----------------+-------------------------------+
```

You can change **SESSIONTIMEZONE** by changing the value of the custom parameter **TIME_ZONE**:

```
ALTER SESSION SET TIME_ZONE='+00:00';
SELECT SESSIONTIMEZONE, ltzcol FROM LocalTZ;
```

The following result is returned:

```
+-----------------+-------------------------------+
| SESSIONTIMEZONE | ltzcol                        |
+-----------------+-------------------------------+
| +00:00          | 25-FEB-20 03:10:08.123000 AM |
+-----------------+-------------------------------+
```

INTERVAL YEAR TO MONTH data type

The INTERVAL YEAR TO MONTH data type stores different objects from the DATE and TIMESTAMP data types. The `TIMESTAMP` data type stores specific timestamps. The `DATE` data type stores specific dates. The `INTERVAL YEAR TO MONTH` data type uses the `YEAR` and `MONTH` elements to store a period of time. This data type can be used to indicate the difference between two date and time values.

Syntax

```
INTERVAL YEAR [(percision)] TO MONTH
```

Parameter

Parameter	Value	Description
percision	0~9	The precision in the `YEAR` element. The default value is 2. That is, if you do not specify this parameter, you can store a maximum interval value of 99 years and 11 months because the maximum interval value cannot exceed 100 years. If you want to store data with the precision that exceeds the default precision of 2 digits, you must explicitly specify this parameter with a value rather than leave this parameter empty.

Date formats

You can use the following formats when you insert a value of the `INTERVAL YEAR TO MONTH` data type. For more information about how to specify the values of interval data types, see Interval literals.

Syntax	Example	Description
INTERVAL 'year-month' YEAR(percision) TO MONTH	INTERVAL '120-3' YEAR(3) TO MONTH	The interval is 120 years and 3 months. Because the value of the `YEAR` element is greater than the default precision of 2, you must set the precision of the `YEAR` element to 3.
INTERVAL 'year' YEAR(percision)	INTERVAL '50' YEAR	The interval is 50 years.
INTERVAL 'month' MONTH	INTERVAL '500' MONTH	The interval is 500 months or 41 years and 8 months.

Examples

Assume that you write the following codes to create three `INTERVAL YEAR TO MONTH` columns named **interval1**, **interval2**, and **interval3** in a table named **Interval_Sample**, and insert numeric values into the columns.

```
CREATE TABLE Interval_Sample (
    interval1 INTERVAL YEAR TO MONTH,
    interval2 INTERVAL YEAR(3) TO MONTH,
    interval3 INTERVAL YEAR TO MONTH
);
INSERT INTO Interval_Sample (interval1, interval2, interval3)
VALUES (INTERVAL '12-3' YEAR TO MONTH, INTERVAL '120-3' YEAR(3) TO MONTH, INTERVAL '40' MONTH);
SELECT * FROM Interval_Sample;
```

Return result:

```
+-----------+-----------+-----------+
| interval1 | interval2 | interval3 |
+-----------+-----------+-----------+
| +12-03    | +120-03   | +03-04    |
+-----------+-----------+-----------+
```

Calculations between interval and other date and time types

ApsaraDB for OceanBase supports the conversion between data types. Therefore, you can perform calculations between interval values and other date values. However, the database allows you to perform addition, subtraction, multiplication, and division between data types by following rules. For more information about the matrix of calculations between date types that are currently supported, see Calculation of DATE and INTERVAL values. For more information about data type conversion, see Data type conversion.

- **Example 1**: When a calculation is performed between interval values, an interval value is returned.

```
SELECT INTERVAL '2-2' YEAR TO MONTH -INTERVAL '1-1' YEAR  TO MONTH calculate1, INTERVAL '2-2' YEAR TO
MONTH + INTERVAL '1-1' YEAR TO MONTH calculate2 FROM DUAL;
```

Return result:

```
+---------------+----------------+
| calculate1    | calculate2     |
+--------------------------------+
| +000000001-01 | +000000003-03  |
+---------------+----------------+
```

- **Example 2**: When a calculation is performed between interval and date and time values, a date value is returned. `SYSDATE` returns the current time **2020-02-27 16:13:50**. The following example returns the date value two months after the current time. The database only supports the format of an interval value plus a date and time value. The format of an interval value minus a date and time value is invalid for the calculation. However, a date time value plus an interval value and a date and time value minus an interval value are valid formats for calculations.

```
SELECT TO_CHAR(INTERVAL '2' MONTH +SYSDATE,'YYYY-MM-DD HH24:MI:SS') calculate3 FROM DUAL;
```

Return result:

```
+---------------------+
| calculate3          |
+---------------------+
| 2020-04-27 16:13:50 |
+---------------------+
```

- **Example 3**: When a calculation is performed between interval and numeric values, an interval value is returned.

Interval values can be multiplied and divided with numeric values. The following example calculates the 2-month interval multiplied by 2 and the 2-day interval divided by 3.

```
SELECT INTERVAL '2' MONTH*2 calculate4, INTERVAL '2' DAY/3 calculate5 FROM DUAL;
```

The intervals of 4 months and 16 hours are returned.

```
+---------------+-------------------------------+
| calculate4    | calculate5                    |
+---------------+-------------------------------+
| +000000000-04 | +000000000 16:00:00.000000000 |
+---------------+-------------------------------+
```

INTERVAL DAY TO SECOND data type

`INTERVAL DAY TO SECOND` stores time periods that are measured in days, hours, minutes, and seconds. This data type is useful for storing a value that represents the precise difference between two date and time values.

Syntax

```
INTERVAL DAY [(precision)] TO SECOND [(fractional_seconds_precision)]
```

Parameters

Parameter	Value	Description
precision	0~9	The precision of the `DAY` element. The default value is 2.
fractional_seconds_precision	0~9	The precision of the fractional part of the `SECOND` element. The default value is 6.

Examples

You can use the following formats when you insert a value of the `INTERVAL DAY TO SECOND` data type. For more information about how to specify values of interval data types, see Interval literals.

Syntax	Examples	Description
INTERVAL 'dd hh:mm:ss' DAY(precision) TO SECOND(fractional_seconds_precision)	INTERVAL '140 5:12:10.2222222' DAY(3) TO SECOND(7)	The interval is 140 days, 5 hours, 12 minutes, and 10.2222222 seconds.
INTERVAL 'dd hh'DAY(precision) TO HOUR	INTERVAL '400 5' DAY(3) TO HOUR	The interval is 400 days and 5 hours.
INTERVAL 'dd hh:mm'DAY(precision) TO MINUTE	INTERVAL '4 5:12' DAY TO MINUTE	The interval is four days, 5 hours, and 12 minutes.
INTERVAL 'hh:mm' HOUR TO MINUTE	INTERVAL '11:20' HOUR TO MINUTE	The interval is 11 hours and 20 minutes.

Syntax	Examples	Description
INTERVAL 'hh:mm:ss' HOUR TO SECOND(fractional_seconds_precision)	INTERVAL '11:12:10.2222222' HOUR TO SECOND(7)	The interval is 11 hours, 12 minutes, and 10.2222222 seconds.
INTERVAL 'dd' DAY(precision)	INTERVAL '14' DAY	The interval is 14 days.
INTERVAL 'hh' HOUR	INTERVAL '160' HOUR	The interval is 160 hours.
INTERVAL 'mm' MINUTE	INTERVAL '14' MINUTE	The interval is 14 minutes.
INTERVAL 'ss' SECOND(fractional_seconds_precision)	INTERVAL '14.666' SECOND(2, 3)	The interval is 14.666 seconds.

The following code can be used to create two `INTERVAL DAY TO SECOND` columns named **interval1** and **interval2** in the **Interval_Sample** table and insert values into these columns.

```
CREATE TABLE Interval_Sample (interval1 INTERVAL DAY TO SECOND, interval2 INTERVAL DAY(3) TO SECOND(3
));
INSERT INTO Interval_Sample (interval1, interval2) VALUES ( INTERVAL '15 06:10:08' DAY TO SECOND, INT
ERVAL '150 06:10:08' DAY(3) TO SECOND(3));
SELECT * FROM Interval_Sample;
```

The following result is returned:

```
+---------------------+-------------------+
| interval1           | interval2         |
+---------------------+-------------------+
| +15 06:10:08.000000 | +150 06:10:08.000 |
+---------------------+-------------------+
```

Calculations on interval data types and other date and time types

ApsaraDB for OceanBase supports the conversion between data types. Therefore, mathematical operations on interval data types and other date and time data types are supported. However, ApsaraDB for OceanBase does not support all addition, subtraction, multiplication, and division operations on two random data types. For more information about the matrix diagram that lists the supported calculations on DATE values, see Calculations on DATE and INTERVAL values. For more information about data type conversion, see Data type conversion.

For more information about examples of calculations on interval data types and other data types, see INTERVAL YEAR TO MONTH data type.

Calculation of DATE and INTERVAL values

You can perform multiple arithmetic operations on the following types of values: dates (`DATE`), timestamps (`TIMESTAMP` , `TIMESTAMP WITH TIME ZONE` , and `TIMESTAMP WITH LOCAL TIME ZONE`), and intervals (`INTERVAL YEAR TO MONTH` and `INTERVAL DAY TO SECOND`).

ApsaraDB for OceanBase calculates results based on the following rules:

- You can use `NUMBER` constants in arithmetic operations that are performed on date and timestamp values instead of interval values. In ApsaraDB for OceanBase, timestamp values are internally converted to date values and `NUMBER` constants in arithmetic datetime and interval expressions are interpreted as the number of days. For example, `SYSDATE` + 1 represents tomorrow. `SYSDATE` - 7 represents a week ago. `SYSDATE` + (10/1440) represents 10 minutes later.

  > **Notice**
  > You cannot multiply or divide date or timestamp values.

- ApsaraDB for OceanBase converts `BINARY_FLOAT` and `BINARY_DOUBLE` operands to `NUMBER` data.
- Each `DATE` value includes a time component and the results of multiple DATE operations include fractions. Each of the fractions represents a portion of a day. For example, 1.5 days is equal to 36 hours. ApsaraDB for OceanBase provides built-in functions to return these fractions for general operations on `DATE` data. For example, the `MONTHS_BETWEEN` function returns the number of months between two dates. The fractional part of the result represents the part of a 31-day month.
- If an operand is a `DATE` or numeric value but contains neither the time zone component nor the fractional second component, the following operations are performed:
  - ApsaraDB for OceanBase converts the other operands to `DATE` data. The exception is the multiplication of a numeric value by an interval. In this case, an interval is returned.
  - If another operand contains a time zone value, ApsaraDB for OceanBase uses the time zone of the current session in the return value.
  - If another operand contains a fractional second value, the fractional second value is lost.
- Assume that you pass timestamps, time intervals, or numeric values to built-in functions that apply to only the `DATE` data type. In this case, ApsaraDB for OceanBase converts non-`DATE` values to `DATE` values.
- When interval calculations return datetime values, the results must be actual datetime values. Otherwise, the database returns errors. For example, errors are returned for the following two statements:

```
SELECT TO_DATE("31-AUG-2004','DD-MON-YYYY") + TO_YMINTERVAL("0-1") FROM DUAL;
SELECT TO_DATE("29-FEB-2004','DD-MON-YYYY") + TO_YMINTERVAL("1-0") FROM DUAL;
```

The first statement fails because the operation of adding a month to a 31-day month results in an invalid date: September 31. The second statement fails because the operation of adding a year to a date that occurs only every four years is invalid. However, the operation of adding four years to February 29 is valid and the calculation result is February 29, 2008:

```
SELECT TO_DATE("29-FEB-2004", "DD-MON-YYYY") + TO_YMINTERVAL("4-0") FROM DUAL;
```

17.1.5.4.1.5. RAW data type

The `RAW` data type stores variable-length binary data. The data is transferred in binary mode across different platforms. The binary data does not need to be converted even if the platforms use different character sets. In ApsaraDB for OceanBase, this data type is used to store binary data or byte strings.

`RAW` is similar to `VARCHAR2`. The RAW data type is declared by using `RAW(length)`. The `length` parameter specifies the value length in bytes. A value in a database column of the RAW data type can be up to 2,000 bytes. A variable of the RAW data type can be up to 2,000 bytes.

Syntax

```
RAW(length)
```

Parameters

Parameter	Description
length	The length of the value. The length is measured in bytes. A value in a database column of the RAW data type can be up to 2,000 bytes. A variable of the RAW data type can be up to 2,000 bytes.

Examples

- **Example 1**: Assume that you declare a column of the `RAW` data type in the **test_raw** table and insert a record into the table.

```
CREATE TABLE test_raw (c1 RAW(10));
INSERT INTO test_raw VALUES (utl_raw.cast_to_raw('1234567890'));
```

Execute the following statement:

```
SELECT utl_raw.cast_to_varchar2(c1) FROM test_raw;
```

The following result is returned:

```
+------------------------------+
|    UTL_RAW.CAST_TO_RAW(C1)   |
+------------------------------+
|          1234567890          |
+------------------------------+
```

- **Example 2**: Assume that you insert two records into the **raw_test** table.

```
CREATE TABLE raw_test (id number, raw_date raw(10));
INSERT INTO raw_test VALUES  (1, hextoraw('ff'));
INSERT INTO raw_test VALUES  (2, utl_raw.cast_to_raw('051'));
```

Execute the following statement:

```
SELECT * FROM raw_test;
```

The following result is returned:

```
+-----------+----------+
|    ID     | RAW_DATE |
+-----------+----------+
|      1    |     ff   |
+-----------+----------+
|      2    |   303531 |
+-----------+----------+
```

The SQL function `HEXTORAW()` converts the data in a string to a hexadecimal value. The SQL function `UTL_RAW.CAST_TO_RAW([VARCHAR2])` stores the ASCII code of each character in a string into a `RAW` field. In this example, **051** is converted to **303531**.

- **Example 3**: ApsaraDB for OceanBase converts a value of the `RAW` type to a value of a character data type. Each character in the result represents the hexadecimal value for four consecutive bits in the value of the `RAW` type. The hexadecimal values can be digits 0 to 9 and letters A to F (or a to f).

For example, the binary number 11001011 of the `RAW` data type is converted to the **CB** string.

To convert character data to a `RAW` value, ApsaraDB for OceanBase interprets each consecutive input character as four consecutive bits of binary data. Then, ApsaraDB for OceanBase builds the resulting `RAW` value by concatenating those bits.

> **Notice**
>
> If an input character is not a hexadecimal value, an error is reported. Hexadecimal values include digits 0 to 9 and letters A to F (or a to f). If the number of characters is odd, the result is undefined.

More information

Character conversion

When data is transferred between databases or between the database character set and the client character set, ApsaraDB for OceanBase automatically converts the `CHAR` and `VARCHAR2` data across database character sets. When the `RAW` data is transferred, ApsaraDB for OceanBase does not convert the characters.

RAW functions

Function	Description
HEXTORAW()	`HEXTORAW()` converts the data in a string to a hexadecimal string. Every two characters in the source string corresponds to 1 byte of the resulting `RAW` value.

Function	Description
RAWTOHEX(rawvalue)	This function converts `rawvalue` of the `RAW` type to a hexadecimal string. Each byte of `rawvalue` is converted to a double-byte string.
UTL_RAW_CAST_TO_RAW([VARCHAR2])	This function only converts the data type from `VARCHAR2` to `RAW`. The content stored in the data remains unchanged.
UTL_RAW_CAST_TO_VARCHAR2([RAW])	This function only converts the data type from `RAW` to `VARCHAR2`. The content stored in the data remains unchanged.
UTL_RAW.BIT_OR(), UTL_RAW.BIT_AND(), and UTL_RAW.BIT_XOR()	Bitwise operation.

17.1.5.4.1.6. Large object data types

Overview of large object data types

The large object (LOB) data types are used to store large amounts of unstructured data, such as texts, images, videos, and spatial data. The LOB data types include binary LOB (`BLOB`) and character LOB (`CLOB`).

When you create a table, you can specify tablespaces and storage characteristics for LOB columns or LOB object attributes. The specified tablespaces and storage characteristics can be different from those specified in the table.

ApsaraDB for OceanBase supports the following LOB data types:

Type	Length	Maximum length (characters)	Character set
BLOB	Variable length	48M	BINARY
CLOB	Variable length	48M	UTF8MB4

BLOB data type

The binary large object (`BLOB`) data type is used to store large amounts of binary objects in a database. You can regard `BLOB` objects as bitstreams that do not have character set semantics. A `BLOB` object has a maximum length of 48 MB and uses the `BINARY` character set.

`BLOB` objects have full transaction support. The changes that you make by using `SQL` statements or the `DBMS_LOB` package participate fully in the transaction. You can submit and roll back operations on `BLOB` values. However, you cannot save a `BLOB` locator in PL/SQL in one transaction and then use it in another transaction or session.

> **Notice**
> If you store an oversized binary file, the database performance is degraded.

In most cases, the `BLOB` fields are used to store the information about large files in a database, such as images, files, and music files. The database converts large files to binary files and then stores the binary files.

In the following example, a table named **blob_table** is created, and the `BLOB` data type is specified for the **blob_cl** column.

```
CREATE TABLE blob_table (blob_cl BLOB);
```

CLOB data type

`CLOB` is short for character large object. The CLOB data type stores single-byte and multi-byte character data. Fixed-width and variable-width character sets are supported. Single-byte and multi-byte character data use the database character set. `CLOB` does not support character sets that have different widths. The maximum length (characters) of bytes that can be stored is 48 MB and the character set is `UTF8MB4` .

`CLOB` objects provide full transactional support. Changes that are made by using `DBMS_ LOB` packages and `SQL` are fully involved in transactions. You can commit and roll back `CLOB` value operations. However, you cannot save a `CLOB` locator to PostgreSQL or SQL Server in one transaction and then use the locator in another transaction or session.

A field of the `VARCHAR2` data type can be a maximum of 32,767 characters in length. If you need to store a field that is more than 32,767 characters in length, you can use the `CLOB` data type. In addition, you can use the `CLOB` data type to store `CHAR` data. For example, XML documents use `CLOB` data to store content.

In the following example, the table named **temp** is created and the data type of the **temp_clob** column is set to `CLOB` :

```
CREATE TABLE temp (temp_clob CLOB);
```

17.1.5.4.2. Comparison rules of data types

17.1.5.4.2.1. Overview of data type comparison rules

The data type comparison rules specify how ApsaraDB for OceanBase compares the values of each data type. The data type comparison rules in ApsaraDB for OceanBase support the following data values:

- Numeric values

- Date values
- Character values

More information

- Data type precedences
- Data type conversions
- Security notes for data conversions

17.1.5.4.2.2. Numeric values

Numeric data includes fixed-point numbers, floating-point numbers, and zeros.

The following list provides the comparison rules for numeric data:

- Larger values are larger than smaller values. For example, 5.5 is larger than 2.1.
- Negative values are smaller than zeros. For example, -3 is smaller than 0 and -200 is smaller than -1.
- Positive values are larger than zeros. For example, 20 is larger than 0 and 100 is larger than 1.

17.1.5.4.2.3. Date values

The `DATE` data type stores date and time information. Each `DATE` value in ApsaraDB for OceanBase stores the following information: year, month, day, hour, minute, and second, but the value does not contain time zone information.

The following list provides the comparison rules for date data:

- The current time is greater than the past time. For example, the date value of May 1, 2018 is greater than the date value of May 1, 2012.
- The time in the afternoon is greater than that in the morning. For example, the date and time value of 15:30:00 on February 2, 2019 is greater than the date and time value of 10:30:00 on February 2, 2019.
- The time in the morning is greater than the time of the previous day. For example, the date and time value of 02:30:00 on March 5, 2019 is greater than the date and time value of 23:30:00 on March 4, 2019.

17.1.5.4.2.4. Character values

Character data is compared by using character values. Character values are compared by using the following two measures:

- Binary and linguistic comparisons
- Blank-padded or nonpadded comparison semantics

Binary and linguistic comparisons

Binary comparison

In the default binary comparison, ApsaraDB for OceanBase compares strings by using the cascaded values of the numeric codes for the characters in the character set of the database. If the numeric value of a character in the character set is larger than the numeric value of another character, the former character is larger. ApsaraDB for OceanBase does not support the American Standard Code for Information Interchange (ASCII) or Extended Binary Coded Decimal Interchange Code (EBCDIC) character set.

Linguistic comparison

In linguistic sorting, both SQL sorting and comparison are performed by using the linguistic rules that are specified by `NLS_SORT`. If the binary sequence of the character encoding does not match the language sequence that is required by the character set, linguistic comparison is used. If the `NLS_SORT` parameter is not set to `BINARY` and the `NLS_COMP` parameter is set to `LINGUISTIC`, language comparison is used.

Blank-padded and nonpadded comparison semantics

Blank-padded comparison semantics

If the blank-padded comparison semantics is used and two values are different in length, ApsaraDB for OceanBase first adds blanks to the end of the shorter value. This ensures that the two values are the same in length. ApsaraDB for OceanBase compares the values character by character until the first different character is identified. The value that has a larger character at the first difference position is considered larger. If the two values do not have different characters, they are considered equal. This rule indicates that the two values are equal only if they are different in the number of trailing blanks.

> **Notice**
>
> ApsaraDB for OceanBase uses the blank-padded comparison semantics only if both values in the comparison are the values of the `CHAR` or `NCHAR` data type, text literals, or the values that are returned by the `USER` function.

Nonpadded comparison semantics

In the blank-padded comparison semantics, ApsaraDB for OceanBase compares two values character by character until the first different character is identified. The value that has a larger character at the position is considered larger. If the two values in different lengths are the same up to the end of the shorter value, the longer value is considered larger. If the two values whose lengths are equal do not have different characters, the two values are considered equal.

> **Notice**
>
> If the data type of one or two values in the comparison is `VARCHAR2` or `NVARCHAR2`, ApsaraDB for OceanBase uses the nonpadded comparison semantics.

Examples

If you compare two character values by using different comparison semantics, the results are different. In the following example, the blank-padded comparison semantics and the nonpadded comparison semantics are used for comparison.

Blank-padded	Nonpadded
'ac' > 'ab'	'ac' > 'ab'
'ab' > 'a '	'ab' > 'a '
'ab' > 'a'	'ab' > 'a'

Blank-padded	Nonpadded
'ab' = 'ab'	'ab' = 'ab'
'ac' > 'ab'	'ac' > 'ab'
'a ' = 'a'	'a ' > 'a'

The results of blank-padded and nonpadded comparisons are generally the same. The comparison example of the last row illustrates the difference between the blank-padded comparison semantics and the nonpadded comparison semantics.

17.1.5.4.2.5. Data type precedence

ApsaraDB for OceanBase uses the data type precedence to specify the order in which data types are implicitly converted.

The following list provides the precedence from highest to lowest for converting data types in ApsaraDB for OceanBase.

1. DATETIME and INTERVAL data types
2. `BINARY_DOUBLE` data type
3. `BINARY_FLOAT` data type
4. `NUMBER` data type
5. Character data types
6. All the other built-in data types

17.1.5.4.2.6. Data type conversion

Generally, an expression cannot contain values of different data types. However, to enable expressions to be calculated, ApsaraDB for OceanBase allows you to implicitly and explicitly convert values from one data type to another.

Implicitly convert data types

If a conversion makes sense, ApsaraDB for OceanBase automatically converts values from one data type to another. Rules of implicitly converting data types are:

- For `INSERT` and `UPDATE` operations, ApsaraDB for OceanBase converts variable values into the column type.
- For `SELECT FROM` operations, ApsaraDB for OceanBase converts the column data type into the destination variable type.
- When character values are compared with numeric values, ApsaraDB for OceanBase converts the character values into numeric values.
- When numeric values are processed, ApsaraDB for OceanBase adjusts the precision and decimal places. The generated numeric data type is different from the numeric data type that is found in the base table.
- Conversions between character values or numeric values and floating-point numbers can be inexact. This is because character types and numeric types use decimal precision to represent numeric values but floating-point numbers use binary precision.

- Assume that a value is converted from `CLOB` to a character type, such as `VARCHAR2`, or from `BLOB` to `RAW`. If the size of data to be converted is larger than the size of the destination data type, the database returns an error.
- When timestamp values are converted into `DATE` values, the fractional second portion of the timestamp values is truncated and rounded.
- The result of converting `BINARY_FLOAT` to `BINARY_DOUBLE` is exact.
- If the number of digits of precision for `BINARY_DOUBLE` exceeds the number of digits supported by `BINARY_FLOAT`, the result of converting `BINARY_DOUBLE` to `BINARY_FLOAT` is inexact.
- When character values are compared with `DATE` values, ApsaraDB for OceanBase converts character data into `DATE` values.
- When a value is assigned, ApsaraDB for OceanBase converts the value on the right side of the equal sign into the data type of the object to which the value is assigned on the left.
- ApsaraDB for OceanBase converts non-character types into character types or national character types during join operations.

Matrix of implicit data type conversions

The following table lists all implicit conversions between data types. You do not need to consider the directions or contexts of conversions. Hyphens (-) indicate that the conversion is not supported.

Data type	CHAR	VARCHAR2	NCHAR	NVARCHAR2	DATE	DATETIME / INTERVAL	NUMBER	BINARY_FLOAT	BINARY_DOUBLE	RAW	CLOB	BLOB
CHAR	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VARCHAR2	Yes	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	-
NCHAR	Yes	Yes	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	-
NVARCHAR2	Yes	Yes	Yes	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	-
DATE	Yes	Yes	Yes	Yes	-	-	-	-	-	-	-	-
DATETIME / INTERVAL	Yes	Yes	Yes	Yes	-	-	-	-	-	-	-	-

NUMBER	Yes	Yes	Yes	Yes	-	-	-	Yes	Yes	-	-	-
BINARY_FLOAT	Yes	Yes	Yes	Yes	-	-	Yes	-	Yes	-	-	-
BINARY_DOUBLE	Yes	Yes	Yes	Yes	-	-	Yes	Yes	-	-	-	-
RAW	Yes	Yes	Yes	Yes	-	Yes [1]	-	-	-	-	Yes	-
CLOB	Yes	Yes	Yes	Yes	-	-	-	-	-	-	-	Yes
BLOB	-	-	-	-	-	-	-	-	-	Yes	-	-

[1] You cannot directly convert from `RAW` to `INTERVAL`. However, you can use `UTL_RAW.CAST_TO_VARCHAR2([RAW])` to convert from `RAW` to `VARCHAR2`. Then, convert the generated `VARCHAR2` value to `INTERVAL`.

Directions of implicit conversions between different character types

Data type	TO_CHAR	TO_VARCHAR2	TO_NCHAR	TO_NVARCHAR2
from CHAR	-	VARCHAR2	NCHAR	NVARCHAR2
from VARCHAR2	VARCHAR2	-	NVARCHAR2	NVARCHAR2
from NCHAR	NCHAR	NCHAR	-	NCHAR2
from NVARCHAR2	NVARCHAR2	NVARCHAR2	NVARCHAR2	-

Examples of implicit data type conversions

Execute the following statement:

```
SELECT 5 * 10 + 'james' FROM DUAL;
```

The statement fails to be executed and you receive the following error message:

```
invalid number
```

This is because ApsaraDB for OceanBase implicitly converts `'james'` into the numeric type, but fails to convert it.

In this example, the string **'2'** is implicitly converted from the `CHAR` data type into the value **2** of the numeric data type. The calculation result is **52**.

Execute the following statement:

```
SELECT 5 * 10 + '2' FROM  DUAL;
```

The following query result is returned:

```
+---------------------+
|     5 * 10 + '2'    |
+---------------------+
|          52         |
+---------------------+
```

Explicitly convert data types

You can use SQL conversion functions to convert data types. SQL functions explicitly convert one data type to another.

Matrix of explicit data type conversions

Data type	To CHAR, VARCHAR2, NCHAR, and NVARCHAR2	To NUMBER	To Datetime/ Interval	To RAW	To CLOB and BLOB	To_BINARY_FLOAT	To_BINARY_DOUBLE
From CHAR, VARCHAR2, NCHAR, and NVARCHAR2	TO_CHAR(char .) and TO_NCHAR(char .)	TO_NUMBER	TO_DATE, TO_TIMESTAMP, TO_TIMESTAMP_TZ, TO_YMINTERVAL, and TO_DSINTERVAL	HEXTORAW	TO_CLOB	TO_BINARY_FLOAT	TO_BINARY_DOUBLE
From NUMBER	TO_CHAR(number) and TO_NCHAR(number)	--	TO_DATE, NUMTOYM_INTERVAL, and NUMTOOLS_INTERVAL	--	--	TO_BINARY_FLOAT	TO_BINARY_DOUBLE
From Datetime/ Interval	TO_CHAR(date) and TO_NCHAR(date)	--	--	--	--	--	--

From RAW	RAWTOHEX and RAWTONHEX	--	--	--	TO_BLOB	--	--
From CLOB and BLOB	TO_CHAR and TO_NCHAR	--	--	--	TO_CLOB	--	--
From BINARY_FLOAT	TO_CHAR(char .) and TO_NCHAR(char .)	TO_NUMBER	--	--	--	TO_BINARY_FLOAT	TO_BINARY_DOUBLE
From BINARY_DOUBLE	TO_CHAR(char .) and TO_NCHAR(char .)	TO_NUMBER	--	--	--	TO_BINARY_FLOAT	TO_BINARY_DOUBLE

Examples of explicit data type conversions

You can use the `TO_CHAR` function to explicitly convert the current time into the desired format for output.

Execute the following statement:

```
SELECT TO_CHAR(SYSDATE, 'YYYY_MM_DD') FROM DUAL;
```

The following query result is returned:

```
+-------------------------------+
| TO_CHAR(SYSDATE,'YYYY_MM_DD') |
+-------------------------------+
| 2020_02_27                    |
+-------------------------------+
```

17.1.5.4.2.7. Security notes for data conversions

If date and time values are implicitly converted to text or explicitly converted to text without specifying a format model, the format model is defined by a global session parameter. The name of the parameter can be `NLS_DATE_FORMAT`, `NLS_TIMESTAMP_FORMAT`, or `NLS_TIMESTAMP_TZ_ FORMAT`. This depends on the type of source data. The values of these parameters can be specified in the client environment or the `ALTER SESSION` statement.

If date and time values in dynamically constructed SQL statements are explicitly converted without specifying a format model, database security is negatively affected when the format model selects a session parameter.

The dynamically constructed SQL statements are those generated by programs or stored procedures. To execute dynamically constructed SQL statements, the built-in procedural language (PL) package `DBMS_SQL` in ApsaraDB for OceanBase is required, or the SQL statement is relevant to the PL statement `EXECUTE IMMEDIATE`. However, these are not the only methods to execute the dynamically constructed SQL text.

In the following example, the data type of `start_date` is `DATE`. The format model that is specified in the session parameter `NLS_DATE_FORMAT` is used to convert the value of `start_date` to text. Then, the result is passed to the SQL text. The date and time format model can consist of text that is enclosed in double quotation marks (").

```
SELECT last_name FROM employees WHERE hire_date > '' || start_date || '';
```

> **Note**
>
> The user who sets globalization parameters for the format model that is used for explicit conversion can determine what text is generated by the previous conversion.
>
> If the SQL statement is executed by a procedure, SQL injection may occur during the execution because the session variables are modified. Procedures that have higher permissions, such as Definer's Rights Procedure, may cause more impact on security.

17.1.5.4.3. Literals

17.1.5.4.3.1. Literal overview

A literal is a notation for representing a fixed value. For most functions and SQL statements, you must specify literals. Literals can also serve as a part of expressions and conditions. ApsaraDB for OceanBase supports the following literals:

- Text literals
- Numeric literals
- Datetime literals
- Interval literals

17.1.5.4.3.2. Text literals

Text literals are strings that are enclosed in single quotation marks `(')` and are used to specify the values of strings in expressions, conditions, SQL functions, and SQL statements.

Text literals have the attributes of the `CHAR` and `VARCHAR2` data types.

- In expressions and conditions, ApsaraDB for OceanBase uses blank-padded comparison semantics for comparison and considers text literals as the `CHAR` data type.
- When you specify text literals, a value of the `CHAR` data type can be up to 2,000 bytes in length. A value of the `VARCHAR2` data type can be up to 32,767 bytes in length.

The following examples show some valid text literals. To show a single quotation mark (') in a string, you must insert a single quotation mark (') before the single quotation mark (') in the string.

```
'Jackie''s raincoat'
'Hello'
'09-MAR-98'
'It's a nice day today'
```

17.1.5.4.3.3. Numeric literals

Numeric literals allow you to use numeric values to specify the fixed number and floating-point number values.

Integer literals

If integers appear in expressions, conditions, SQL functions, and SQL statements, you must use integer literals to specify values.

The following integer literals are valid:

```
8
+186
-15
```

Number literals and floating-point literals

If numbers appear in expressions, conditions, SQL functions, and SQL statements, you must use number or floating-point literals to specify values.

The following number literals are valid:

```
12
+6.87
0.5
25e-03
-9
```

The following floating-point literals are valid:

```
25f
+6.34F
0.5d
-1D
```

A number literal can store a number whose maximum precision is 38 digits. If a literal requires a higher precision than that provided by `NUMBER`, `BINARY_FLOAT`, or `BINARY_DOUBLE`, ApsaraDB for OceanBase truncates the value. If the range of the literal exceeds the range that is supported by `NUMBER` `BINARY_FLOAT`, or `BINARY_DOUBLE`, ApsaraDB for OceanBase reports an error.

Examples

The decimal point delimiter in a numeric literal is always a dot (.). If you specify a text literal in a position where a numeric literal is expected, the text literal is converted to a numeric literal.

In the following example, **2** is multiplied by the numeric literal **2.2** and **2** is multiplied by the text literal **'3.3'**:

```
SELECT 2*2.2, 2*'3.3' FROM DUAL;
```

The following result is returned:

```
+-------+---------+
| 2*2.2 | 2*'3.3' |
+-------+---------+
|   4.4 |     6.6 |
+-------+---------+
```

17.1.5.4.3.4. Datetime literals

Date literals

You can use a string to specify a date literal. You can also use a `TO_DATE` function to convert a character or numeric value to a date value. You can use a `TO_DATE` expression in place of a string to specify a date literal. This applies to only date literals.

```
TO_DATE('2020-03-25 11:05:00', 'YYYY-MM-DD HH24:MI:SS')
```

When you use a date value to specify a date literal, you must use the date value of the Gregorian calendar. You can also use the ANSI format to specify a date literal, as shown in the following example. The ANSI date literal does not contain the time information and must be specified in the **YYYY-MM-DD** format.

```
DATE '2020-03-25'
```

You can also use the default date value of the database to specify a date literal. If the default value is used in a date expression, ApsaraDB for OceanBase automatically converts the character value in the default date format to a date value. The default date value of the database is specified by the initialization parameter `NLS_DATE_FORMAT`. In this example, the default format is **DD-MON-RR**.

```
TO_DATE('25-FEB-20', 'DD-MON-RR')
```

If you specify a date value that does not include the time part, the default time is midnight (00:00:00 in the 24-hour clock and 12:00:00 in the 12-hour clock). If a specified date value does not include the date part, the default date is the first day of the current month.

The `DATE` column of a table in ApsaraDB for OceanBase always contains both the date and time fields. Therefore, if you query the `DATE` column, you must specify the time field in the query or make sure that the time field in the `DATE` column is specified as midnight. Otherwise, the database may not return the query result that you expect. For example, create a **Date_Literals** table that contains the **id** column and the date column **datecol**.

```
CREATE TABLE Date_Literals (id NUMBER, datecol DATE);
```

Insert the system date and time **SYSDATE** of the current session into the table. In this example, the `TRUNC` function is used to set the time field to midnight. The `TRUNC` function truncates the date part of **SYSDATE**. This way, the time in the **datecol** column is automatically filled with the default midnight time:

```
INSERT INTO Date_Literals VALUES (1,SYSDATE);
INSERT INTO Date_Literals VALUES (2,TRUNC(SYSDATE));
```

The following data in the table is available:

```
+------+---------------------+
| id   | datecol             |
+------+---------------------+
|    1 | 25-FEB-20 11:28:16  |
|    2 | 25-FEB-20 00:00:00  |
+------+---------------------+
```

If the query does not contain the time information, you can use the greater than or less than condition instead of the equal to or not equal to condition in the query:

```
SELECT * FROM Date_Literals WHERE datecol > TO_DATE('2020-02-24', 'YYYY-MM-DD');
```

The following result is returned:

```
+------+---------------------+
| id   | datecol             |
+------+---------------------+
|    1 | 25-FEB-20 11:28:16  |
|    2 | 25-FEB-20 00:00:00  |
+------+---------------------+
```

If you use the equal to condition, only the date whose time information is midnight is returned because the query does not contain the time information.

```
SELECT * FROM Date_Literals WHERE datecol = TO_DATE('2020-02-25', 'YYYY-MM-DD');
```

The following result is returned:

```
+------+---------------------+
| id   | datecol             |
+------+---------------------+
|    2 | 25-FEB-20 00:00:00  |
+------+---------------------+
```

You can also filter out the time field in the **datecol** column and query only the date field:

```
SELECT * FROM Date_Literals WHERE TRUNC(datecol) = DATE '2020-02-25';
```

The following result is returned:

```
+------+---------------------+
| id   | detacol             |
+------+---------------------+
|    1 | 25-FEB-20 11:28:16  |
|    2 | 25-FEB-20 00:00:00  |
+------+---------------------+
```

Timestamp literals

ApsaraDB for OceanBase supports the following three types of timestamp literals:

- TIMESTAMP
- TIMESTAMP WITH TIME ZONE
- TIMESTAMP WITH LOCAL TIME ZONE

TIMESTAMP literals

The `TIMESTAMP[(scale)]` data type stores year, month, day, hour, minute, and second, and fractional second values. When you specify a TIMESTAMP literal, you can set the maximum precision of the second field to the nanosecond that lies in the ninth digit.

```
TIMESTAMP '2020-02-25 11:26:18.316'
```

TIMESTAMP WITH TIME ZONE literals

TIMESTAMP WITH TIME ZONE literals are timestamp literals that include time zone information.

The `TIMESTAMP [(scale)] WITH TIME ZONE` data type is a variant of the `TIMESTAMP[(scale)]` data type. The TIMESTAMP [(scale)] WITH TIME ZONE data type stores the information, such as time zone offsets and the names of time zone regions (TZRs), in addition to the information that the `TIMESTAMP[(scale)]` data type stores.

When you specify a TIMESTAMP WITH TIME ZONE literal, you must specify the time zone information. You can also set the maximum precision of the second field to the nanosecond that lies in the ninth digit. In the following example, the time zone offset is used to specify the value of the time zone field.

```
TIMESTAMP '2020-02-25 11:26:18.316 +08:00'
```

Assume that the values of two TIMESTAMP WITH TIME ZONE literals represent the same point in time in the Greenwich Mean Time (GMT) time zone. The two literals are considered as the same literals even if the values of the time zone fields for these two literals are different. In the following example, 08:00:00 in the GMT-8 time zone and 11:00:00 in the GMT-5 time zone are actually the same point in time:

```
TIMESTAMP '2020-04-25 08:26:18.316 -08:00'
TIMESTAMP '2020-04-25 11:26:18.316 -05:00'
```

In a literal, we can replace the time zone offset with the name of a TZR. In the following example, **-08:00** is replaced with **America/Los_Angeles**.

```
TIMESTAMP '2020-02-01 11:00:00 America/Los_Angeles'
```

In some regions, the daylight saving time (DST) switches. To eliminate the time ambiguity when the DST switches, you can use both the TZR name and a corresponding abbreviation TZD. This ensures that the literal value is the daylight saving time.

```
TIMESTAMP '2020-06-01 11:00:00 America/Los_Angeles PDT'
```

TIMESTAMP WITH LOCAL TIME ZONE literals

`TIMESTAMP [(scale)] WITH LOCAL TIME ZONE` is a data type that contains the information about the local time zone. In ApsaraDB for OceanBase, specific TIMESTAMP WITH LOCAL TIME ZONE literals are unavailable. The `TIMESTAMP [(scale)] WITH LOCAL TIME ZONE` literals are assigned values by using other valid date and time literals. The following table shows some formats that can be used to insert values into the TIMESTAMP WITH LOCAL TIME ZONE column, and the return values of queries:

Value specified in the INSERT statement	Return value of the query
'25-FEB-20'	25-FEB-20 00.00.000000
SYSTIMESTAMP	25-FEB-20 14:28:41.264258
TO_TIMESTAMP('25-FEB-2020', 'DD-MON-YYYY')	25-FEB-20 00.00.000000
SYSDATE	25-FEB-20 02.55.29.000000 PM

Value specified in the INSERT statement	Return value of the query
TO_DATE('25-FEB-20', 'DD-MON-YYYY')	25-FEB-20 12.00.00.000000 AM
TIMESTAMP'2020-02-25 8:00:00 America/Los_Angeles'	25-FEB-20 08.00.00.000000 AM

17.1.5.4.3.5. Interval literals

An interval literal specifies the value of a period of time. ApsaraDB for OceanBase supports the following two types of interval literals:

- INTERVAL YEAR TO MONTH
- INTERVA DAY TO SECOND

Leading and trailing fields

Each type of interval literal contains a leading field and an optional trailing field. The leading field defines the basic unit of measured date or time. The trailing field defines the smallest increment of the considered basic unit. For example, DAY TO MINUTE is used to specify an interval literal whose minimum unit is month. The leading field is `YEAR` and the trailing field is `MINUTE`. The trailing field is optional. When you specify the interval literal, the trailing field can be omitted.

Interval literals have the following fields: `YEAR`, `MONTH`, `DAY`, `HOUR`, `MINUTE`, and `SECOND`. The weights of these fields are in descending order. The weight of the `YEAR` field is the largest. If you need to specify a trailing field, the weight of the trailing field in the literal must be lower than that of the leading field. Otherwise, the specified trailing field is invalid. For example, `INTERVAL '1-2' DAY TO YEAR` is an invalid literal.

The number of digits in the value of the leading field ranges from 0 to 9. The default value is 2. The `SECOND` field specifies the number of seconds. This field value can be accurate to at most nine decimal places and at least zero decimal place. The default precision is six decimal places. When a field value exceeds the specified range, the database returns an error. If the number of decimal places of the `SECOND` field exceeds the specified precision, the field value is rounded to the value that is a specified precision.

INTERVAL YEAR TO MONTH literals

The `INTERVAL YEAR TO MONTH` literal specifies a period of time in the unit of year and month.

The following examples show you some `INTERVAL YEAR TO MONTH` literals:

Example	Description
INTERVAL '265-2' YEAR(3) TO MONTH	An interval of 265 years and two months. The precision of the leading field `YEAR` is greater than the default precision: two digits. The specified precision must match the number of digits of the value.
INTERVAL '265' YEAR(3)	An interval of 265 years.

Example	Description
INTERVAL '500' MONTH(3)	An interval of 500 months or an interval of 41 years and eight months.
INTERVAL '10' MONTH	An interval of 10 months.
INTERVAL '123' YEAR	Returns an error. The value 123 exceeds the default precision: two digits.

You can add one `INTERVAL YEAR TO MONTH` literal to another `INTERVAL YEAR TO MONTH` literal or subtract one INTERVAL YEAR TO MONTH literal from another INTERVAL YEAR TO MONTH literal. For example: `INTERVAL '6-2' YEAR TO MONTH` plus `INTERVAL'21' MONTH` equals to `INTERVAL '7-11' YEAR TO MONTH`.

INTERVAL DAY TO SECOND literals

The `INTERVAL DAY TO SECOND` literal specifies a period of time for which the day and the specific time are used as the unit.

The following examples show some `INTERVAL DAY TO SECOND` literals:

Example	Description
INTERVAL '4 5:12:10.222' DAY TO SECOND(3)	An interval of four days, 5 hours, 12 minutes, and 10.222 seconds. The default decimal point precision of the `SECOND` field is 6. If you do not manually set the precision to 3, the system pads the decimal places of the returned result with zeros.
INTERVAL '4 5:12' DAY TO MINUTE	An interval of four days, 5 hours, and 12 minutes.
INTERVAL '400 5' DAY(3) TO HOUR	An interval of 400 days and 5 hours. The precision of the leading field `DAY` exceeds the default precision: two digits. Manually specify the precision as 3.
INTERVAL '400' DAY(3)	An interval of 400 days.
INTERVAL '11:12:10.2222222' HOUR TO SECOND(7)	An interval of 11 hours, 12 minutes, and 10.2222222 seconds. The precision for the value of the `SECOND` field value exceeds the default precision: six digits. The precision that matches the value is manually specified here.

Example	Description
INTERVAL '11:20' HOUR TO MINUTE	An interval of 11 hours and 20 minutes.
INTERVAL '10' HOUR	An interval of 10 hours.
INTERVAL '10:22' MINUTE TO SECOND	An interval of 10 minutes and 22 seconds.
INTERVAL '10' MINUTE	An interval of 10 minutes.
INTERVAL '4' DAY	An interval of four days.
INTERVAL '25' HOUR	Indicates an interval of 25 hours.
INTERVAL '40' MINUTE	An interval of 40 minutes.
INTERVAL '120' HOUR(3)	An interval of 120 hours.
INTERVAL '30.12345' SECOND(2,4)	An interval of 30.1235 seconds. The number of decimal places of the SECOND value exceeds the specified precision. Therefore, this value is rounded to the fourth decimal place.

You can perform an addition or subtraction operation on an `INTERVAL DAY TO SECOND` literal and another `INTERVAL DAY TO SECOND` literal. For example, `INTERVAL'20' DAY` minus `INTERVAL'239' HOUR` is equal to `INTERVAL'10-1' DAY TO SECOND`.

17.1.5.4.4. Formatting

17.1.5.4.4.1. Formatting overview

Formatting specifies the format of datetime or numeric data that is stored in databases. When you convert a string to a datetime or numeric value, the formatting determines how ApsaraDB for OceanBase converts and stores the string. In an SQL statement, you can set the parameters of the `TO_CHAR`, `TO_NUMBER`, and `TO_DATE` functions to specify the following items:

- The format of the values that are returned by ApsaraDB for OceanBase
- The format of the values that are stored in ApsaraDB for OceanBase

ApsaraDB for OceanBase supports the following types of data formatting:

Number formatting

Number formatting specifies the formats of fixed-point and floating-point numbers that are stored in databases. When you need to convert `NUMBER`, `BINARY_FLOAT`, or `BINARY_ DOUBLE` values in an SQL statement to `VARCHAR2` values, you can use number formatting in functions. A number formatting model consists of one or more number format elements. For more information, see Number formatting.

Datetime formatting

Datetime formatting specifies the format of date and time data that is stored in databases. The total length of a string that is obtained after datetime formatting cannot exceed 22 characters. When you need to convert character values in a non-default format to values in the datetime format, you can use datetime formatting in functions. A datetime formatting model consists of one or more datetime format elements. For more information, see Datetime formatting. For more information about the rules for converting strings to date values, see String-to-date conversion rules.

The `RR` datetime format element is similar to the `YY` datetime format element. However, the RR element offers additional flexibility for storing the values of dates that are not covered by the current century. For more information about RR, see RR datetime format element.

Formatting modifiers

ApsaraDB for OceanBase does not support `FX` or `FM` formatting modifiers.

17.1.5.4.4.2. Number formatting

Number formatting specifies the formats of fixed-point and floating-point numbers that are stored in databases.

Number formatting in functions

The following functions for numeric type conversion use number formatting:

- Assume that `NUMBER`, `BINARY_FLOAT`, or `BINARY_ DOUBLE` values appear in expressions, conditions, SQL functions, and SQL statements. If you need to convert these values to `VARCHAR2` values, set the corresponding parameters of the `TO_CHAR` function to specify the formats of these numeric values.
- Assume that `CHAR` or `VARCHAR2` values appear in expressions, conditions, SQL functions, and SQL statements. If you need to convert these values to `NUMBER` values, set the corresponding parameters of the `TO_NUMBER` function to specify the formats of these numeric values. The `NLS_NUMERIC_CHARACTERS` parameter is not supported. If you need to convert the values to `BINARY_FLOAT` or `BINARY_DOUBLE` values, set the corresponding parameters of `TO_BINARY_FLOAT` and `TO_BINARY_DOUBLE` functions to specify the formats of these numeric values.

Number formatting rounds a value to the specified number of significant digits. If a value has a decimal part and the number of significant digits to the left is greater than that specified in the format, the value is replaced with `#`. If the positive value of the `NUMBER` type is extremely large and cannot be represented in the specified format, the value is replaced with the infinity symbol (~). If the negative value of the `NUMBER` type is extremely small and cannot be represented in the specified format, the value is replaced with the negative infinity symbol (-~).

Number format elements

ApsaraDB for OceanBase is different from Oracle databases because the number format elements in ApsaraDB for OceanBase support only standard numeric formats. The following table lists the number format elements that are supported by ApsaraDB for OceanBase.

Element	Example	Description
. (decimal point)	99.99	Returns a decimal number where the decimal point is in the specified place. **Constraint**: In number formatting, you can specify only one decimal point.
0	0999 9990	For 0999, leading zero values are returned. For 9990, trailing zero values are returned.
9	9999	Returns a value that has the specified number of digits. If the value is positive, a number with leading white-space characters is returned. If the value is negative, a number with a leading minus sign (-) is returned. Leading zeros are not displayed and are replaced with white-space characters for all of the numeric values except zero values. A zero is returned for each non-leading zero value that is included in the integer part of the fixed-point number.

If the format parameter is omitted, ApsaraDB for OceanBase converts the numeric value to a `VARCHAR2` value of a sufficient length to retain all of the significant digits of the value.

Examples

Execute the following statement:

```
SELECT TO_CHAR(0, '99.99') FROM DUAL;
```

The following query result is returned:

```
+--------------------+
| TO_CHAR(0,'99.99') |
+--------------------+
|    .00             |
+--------------------+
```

The following table lists the results of queries where **number** is set to different values and the 'fmt' format element is used.

```
SELECT TO_CHAR(number, 'fmt')  FROM DUAL;
```

number	'fmt'	Result
0	99.99	' .00'
+0.1	99.99	' .10'
-0.2	99.99	' -.20'
0	90.99	' 0.00'
+0.1	90.99	' 0.10'
-0.2	90.99	' -0.20'
0	9999	' 0'
1	9999	' 1'
+123.456	999.999	' 123.456'
-123.456	999.999	'-123.456'

When the 'fmt' parameter is omitted, the following statement is executed:

```
SELECT TO_CHAR(123.456) FROM DUAL;
```

The following result is displayed:

```
+------------------+
| TO_CHAR(123.456) |
+------------------+
| 123.456          |
+------------------+
```

When the `TO_CHAR` result is more than 40 bytes in length or the input parameter is of the BINARY_DOUBLE or BINARY_FLOAT type, the result is in the scientific notation format.

Example:

```
SELECT TO_CHAR(12355555555555555555555555555555555555555555555555555) FROM DUAL;
```

The following result is displayed:

```
+-----------------------------------------------------------------+
| TO_CHAR(123555555555555555555555555555555555555555555555555) |
+-----------------------------------------------------------------+
| 1.235555555555555555555555555555556E+49                         |
+-----------------------------------------------------------------+
```

17.1.5.4.4.3. Datetime formatting

Datetime formatting specifies the format of date and time data that is stored in databases. The total length of a string that is obtained after datetime formatting cannot exceed 22 characters.

Datetime formatting in functions

Datetime formatting appears in the following data type conversions:

- When you use the `TO_DATE`, `TO_TIMESTAMP`, or `TO_TIMESTAMP_TZ` function to convert a character value in a non-default format to a datetime value, specify the datetime format. To specify the datetime format, set the corresponding parameter of the function that you use.
- When you use the `TO_CHAR` function to convert a datetime value to a character value in a non-default format, set the corresponding parameter of the function to specify the format.

You can use the following methods to specify the datetime format:

- Explicitly specify the datetime format by using the `NLS_DATE_FORMAT`, `NLS_TIMESTAMP_FORMAT`, or `NLS_TIMESTAMP_TZ_FORMAT` session parameter.
- Implicitly specify the datetime format by using the `NLS_TERRITORY` session parameter
- Change the default datetime format of a session by executing the `ALTER SESSION` statement.

Datetime formatting

A datetime formatting model consists of one or more datetime format elements. For more information about the format elements that are supported by ApsaraDB for OceanBase, see the table of datetime format elements.

- In a formatting string, the same format element cannot appear twice, and format elements that represent similar information cannot be combined. For example, you cannot use `SYYYY` and `BC` elements in the same formatting string.
- All of the format elements can be used in `TO_CHAR`, `TO_DATE`, `TO_TIMESTAMP`, and `TO_TIMESTAMP_TZ` functions.
- The following datetime format elements can be used in timestamp and interval formatting, but not in `DATE` formatting: `FF`, `TZD`, `TZH`, `TZM`, and `TZR`.
- Many datetime format elements are padded with blank spaces or zero values to a specified length.

> **Notice**
>
> In ApsaraDB for OceanBase, we recommend that you use the four-digit year element (`YYYY`). The year element with a shorter length affects query optimization because the year can be determined only at the runtime.

Table of datetime format elements

Element	Supported by datetime functions	Description
- / , . ; : "text"	Yes	Punctuations and quoted text are copied to the result.
AD A.D.	Yes	The anno domini (A.D.) year. The periods (.) can be retained or removed.
AMA.M.	Yes	The morning. The periods (.) can be retained or removed.
BCB.C.	Yes	The before Christ (B.C.) year. The periods (.) can be retained or removed.
D	Yes	The day (1 to 7) of the week.
DAY	Yes	The name of the day.
DD	Yes	The day (1 to 31) of the month.
DDD	Yes	The day (1 to 366) of the year.
DL	Yes	Date and time data is printed only in the fixed format. For example, "Monday, January, 01, 1996" is in the fixed format.
DS	Yes	Date and time data is printed only in the fixed format. For example, "10-10-1996" is in the fixed format.
DY	Yes	The abbreviated name of the date. The day of the week is returned.

Element	Supported by datetime functions	Description
FF [1..9]	Yes	The fractional seconds. Use the numbers 1 to 9 to specify the number of digits in the fractional second portion of the return value. By default, the precision that is specified by the datetime data type is used. This element is valid in timestamp and interval formatting, but invalid in `DATE` formatting.
FX	Yes	Requires an exact match between character data and the format model.
HHHH12	Yes	The hour (1 to 12). The 12-hour clock is used.
HH24	Yes	The hour (0 to 23) The 24-hour clock is used.
YYYY	Yes	The year in the four-digit format.
MI	Yes	The minute (0 to 59)
MM	Yes	The month (01 to 12). The value 01 represents January.
MON	Yes	The abbreviated name of the month.
MONTH	Yes	The name of the month.
PMP.M.	Yes	The afternoon. The periods (.) can be retained or removed.
Q	Yes	The quarter (1, 2, 3, and 4). The first quarter lasts from January to March.
RR	Yes	The year in the two-digit format.

Element	Supported by datetime functions	Description
RRRR	Yes	The year. Four-digit or two-digit inputs can be entered.
SS	Yes	The second (0 to 59).
SSSSS	Yes	The number of seconds (0 to 86400) after midnight.
TZD	Yes	The daylight saving time (DST) information. The TZD value is an abbreviated time zone string that contains the DST information. This element is valid in timestamp and interval formatting, but invalid in `DATE` formatting.
TZH	Yes	The time zone hour. This element is valid in timestamp and interval formatting, but invalid in `DATE` formatting.
TZM	Yes	The time zone minute. This element is valid in timestamp and interval formatting, but invalid in `DATE` formatting.
TZR	Yes	The region information about the time zone. This element is valid in timestamp and interval formatting, but invalid in `DATE` formatting.
X	Yes	The decimal point, which is always a period (.).
Y,YYY	Yes	The year with a comma (,).
YYYYSYYYY	Yes	The year in the four-digit format. S means that a minus sign (-) is used to represent a B.C. date.
YYYYYY	Yes	The last one, two, or three digits of the year.

> **Note**
>
> Datetime functions are `TO_CHAR`, `TO_DATE`, `TO_TIMESTAMP`, and `TO_TIMESTAMP_TZ`.

Take note of the following point: In the preceding conversions, the input date string must match the format elements. Otherwise, an error is returned. The following example is provided:

```
SELECT TO_DATE( '31 Aug 2020', 'DD MON YYYY' ) FROM DUAL;

+-----------------------------------+
| TO_DATE('31AUG2020','DDMONYYYY') |
+-----------------------------------+
| 2020-08-31 00:00:00               |
+-----------------------------------+
```

If some elements are missing from your formatting string, the system returns an error:

```
SELECT TO_DATE( '31 Aug 2020', 'DD MON YYY' ) FROM DUAL;

ORA-01830: date format picture ends before converting entire input string
```

Uppercase letters in date format elements

Uppercase letters in spelled-out words, abbreviations, or Roman numerals are also capitalized in the corresponding format elements. For example, the `DAY` date format element generates `MONDAY` where every letter is capitalized. `Day` generates `Monday` that is in the same format as the Day element. `day` generates `monday` that is in the same format as the day element.

```
OceanBase(SYS@SYS)>SELECT TO_CHAR(sysdate,'mon')  AS nowMonth FROM DUAL;
+-----------+
| NOWMONTH |
+-----------+
| sep      |
+-----------+

OceanBase(SYS@SYS)>SELECT TO_CHAR(sysdate,'MON')  AS nowMonth FROM DUAL;
+-----------+
| NOWMONTH |
+-----------+
| SEP      |
+-----------+
```

Punctuations and character literals in datetime formatting

Date formatting is required for the following characters. These characters appear in the return value in the same positions as they appear in the formatting string:

- Punctuations, such as hyphens (-), slashes (/), commas (,), periods (.), and colons (:)
- Character literals, which are enclosed in double quotation marks (")

ApsaraDB for OceanBase can convert strings to dates based on your business needs. If you use the `TO_DATE` function and each numeric element of the input string contains the maximum number of digits allowed for formatting, the formatting string matches the input string.

- **Example 1**: In the **MM/YY** format element, **02** corresponds to **MM** and **07** corresponds to **YY**.

Execute the following statement:

```
SELECT TO_CHAR(TO_DATE('0207','MM/YY'),'MM/YY') FROM DUAL;
```

The following query result is returned:

```
+-----------------------------------------+
| TO_CHAR(TO_DATE('0207','MM/YY'),'MM/YY') |
+-----------------------------------------+
| 02/07                                   |
+-----------------------------------------+
```

- **Example 2**: ApsaraDB for OceanBase allows matching between punctuation characters in formatting and non-alphanumeric characters. For example, **#** corresponds to **/**.

Execute the following statement:

```
 SELECT TO_CHAR (TO_DATE('02#07','MM/YY'), 'MM/YY') FROM DUAL;
```

The following query result is returned:

```
+-----------------------------------------+
| TO_CHAR(TO_DATE('02#07','MM/YY'),'MM/YY') |
+-----------------------------------------+
| 02/07                                   |
+-----------------------------------------+
```

Date format elements and globalization support

In ApsaraDB for OceanBase, you can use the `NLS_DATE_LANGUAGE` and `NLS_LANGUAGE` parameters to specify the language for datetime format elements. The default value is `AMERICAN`. The value cannot be changed. Therefore, globalization is not supported.

Example: By default, the language for datetime format elements is `American`. The other languages are not supported.

```
SELECT TO_CHAR (SYSDATE, 'DD/MON/YYYY', 'nls_date_language=''Traditional Chinese'' ') FROM DUAL;
```

An error is returned in the query result because the specified value for the language parameter is not supported.

```
ERROR-12702: invalid NLS parameter string used in SQL function
```

More information

- String-to-date conversion rules

17.1.5.4.4.4. RR datetime format element

The `RR` datetime format element is similar to the `YY` datetime format element. However, the RR element offers additional flexibility for storing the values of dates that are not covered by the current century. In the `YY` datetime format element, you must specify all of the digits of the year. In the `RR` datetime format element, you need only to specify the last two digits of the year to store a date value.

Assume that the `RR` datetime format element is used in conjunction with the `TO_DATE` function. In this case, the century of the return value varies based on the specified two-digit year and the last two digits of the current year. If the `YY` datetime format element is used in conjunction with the `TO_DATE` function, the returned year always has the same first two digits as the current year.

Assume that the specified two-digit year ranges from 00 to 49. In this case, if the last two digits of the current year ranges from 00 to 49, the returned year has the same first two digits as the current year. If the last two digits of the current year ranges from 50 to 99, the first two digits of the returned year indicate the next century.

Assume that the specified two-digit year ranges from 50 to 99. In this case, if the last two digits of the current year ranges from 00 to 49, the first two digits of the returned year indicate the previous century. The previous century is the one before the century of the current year. If the last two digits of the current year ranges from 50 to 99, the returned year has the same first two digits as the current year.

The `RR` datetime format element returns the same value for the years whose first two digits are different, as shown in the following examples: Assume that these queries were run during the period from 1950 to 1999. Execute the following statement:

```
SELECT TO_CHAR(TO_DATE('27-OCT-98', 'DD-MON-RR'), 'YYYY') "Year1" ,
TO_CHAR(TO_DATE('27-OCT-17', 'DD-MON-RR'), 'YYYY') "Year2"  FROM DUAL;
```

The following query result is returned:

```
+-------+-------+
| Year1 | Year2 |
+-------+-------+
| 2017  | 1998  |
+-------+-------+
```

Assume that these queries were run during the period from 2000 to 2049. Execute the following statement:

```
SELECT TO_CHAR(TO_DATE('27-OCT-98', 'DD-MON-RR'), 'YYYY') "Year1" ,
TO_CHAR(TO_DATE('27-OCT-17', 'DD-MON-RR'), 'YYYY') "Year2"  FROM DUAL;
```

The following query result is returned:

```
+-------+-------+
| Year1 | Year2 |
+-------+-------+
| 2017  | 1998  |
+-------+-------+
```

Notice

The same value is returned regardless of whether queries were run before or after 2000.

17.1.5.4.4.5. String-to-date conversion rules

The following conversion rules govern the conversion from a string value to a date value:

- If you specify all numeric values for numeric format elements, including leading zeros, you can omit punctuation marks contained in the format string from the date string. Specify 02 instead of 2 for the two-digit format elements such as `MM`, `DD`, and `YY`.
- You can omit time fields found at the end of the format string from the date string.
- You can use any non-alphanumeric character in the date string to match the punctuation mark in the format string.

17.1.5.4.5. Null values

17.1.5.4.5.1. Null value overview

Null values are invalid, unspecified, unknown, or unpredictable values in database tables. The occurrences of null values are not restricted by the `NOT NULL` or `PRIMARY KEY` constraint. The result of each arithmetic expression that contains `NULL` is `NULL`.

ApsaraDB for OceanBase supports the following three types of null values:

Null values in SQL functions

The null values of this type are the null values of the parameters in SQL functions. When the parameters of SQL functions have null values, most scalar functions return `NULL` and analytic functions ignore null values. Null values in SQL functions are divided into two types. The following table describes the two types.

Null value	Description
Null values in NVL functions	If `expr1` in the `NVL(expr1,expr2)` expression is not NULL, the expression returns `expr1`. Otherwise, the expression returns `expr2`.
Null values in analytic functions	When an analytic function such as `AVG`, `MAX`, `SUM`, or `COUNT` is used, NULL records are ignored.

Null values in comparison conditions

The null values of this type are the `NULL` values that are found in comparison conditions and used for comparison. Only `IS NULL` and `IS NOT NULL` comparators can be used to test for null values. `NULL` is incomparable to other values because it indicates that data is missing. This means that `NULL` cannot be equal to, unequal to, greater than, or smaller than another numeric value or another null value.

Null values in conditional expressions

Null values in conditional expressions are the NULL values in the `= NULL`, `! = NULL`, `NULL =`, and `NULL ! =` conditions. These `NULL` values are used for logical evaluation. If conditions evaluate to `UNKNOWN`, no rows are returned.

17.1.5.4.5.2. Null values in SQL functions

Null values in SQL functions refer to the null arguments in the functions. If you pass null arguments to SQL functions, most scalar functions return `NULL`, and analytic functions ignore the null arguments. You can determine whether a null value occurs based on the return value of the `NVL` function.

Null value in the NVL function

`NVL(expr1,expr2)` is the expression of the `NVL` function. If `expr1` is not `NULL`, `expr1` is returned. Otherwise, `expr2.` is returned.

In the following example, the statement is used to query the return value of the `NVL(expr1,0)` expression when the `expr1` parameter is set to `NULL`.

Execute the following statement:

```
SELECT NVL(NULL,0) FROM DUAL;
```

The following result is returned:

```
+-------------+
| NVL(NULL,0) |
+-------------+
|           0 |
+-------------+
```

If `expr1` is `NULL`, the `NVL(expr1,0)` expression returns 0. If `expr1` is not `NULL`, the expression returns `NULL`.

Null values in analytic functions

When you use analytic functions such as `AVG`, `MAX`, `SUM`, and `COUNT`, `NULL` records are ignored.

In the following example, the data is inserted into the **tbl_a** table, and the following statement is executed:

```
CREATE TABLE tbl_a (col_a varchar2(1),  col_b int );
INSERT INTO tbl_a VALUES (NULL, 3);
INSERT INTO tbl_a VALUES (NULL, NULL);
INSERT INTO tbl_a VALUES (NULL, 1);
```

Execute the following statement:

```
SELECT * FROM tb1_a;
```

The following result is returned:

```
+-------+-------+
| COL_A | COL_B |
+-------+-------+
|  NULL |   3   |
+-------+-------+
|  NULL |  NULL |
+-------+-------+
|  NULL |   1   |
+-------+-------+
```

The following result is returned:

```
SELECT AVG(col_b) FROM tbl_a; -- The result is 2.
SELECT MAX(col_b) FROM tbl_a; -- The result is 3.
SELECT SUM(col_b) FROM tbl_a; -- The result is 4.
SELECT COUNT(col_b) FROM tbl_a; -- The result is 2.
SELECT COUNT(col_a) FROM tbl_a; -- The result is 0.
SELECT COUNT(*) FROM tbl_a; -- The result is 3.
```

`NULL` records are ignored.

17.1.5.4.5.3. Null values in comparison conditions

Null values in comparison conditions are `NULL` compared to other conditions. To test for nulls, you can use only the comparison operators `IS NULL` and `IS NOT NULL`. `NULL` is not comparable to other values because it indicates a lack of data. That is, `NULL` cannot be compared to other values, including null values, by using "equal to", "not equal to", "greater than", or "smaller than".

In addition, ApsaraDB for OceanBase considers two null values to be equal when it performs calculations by using the `DECODE` function. Two null values are also considered to be equal if they appear in compound keys.

The following example shows that you judge the results of comparison conditions based on the values of A:

Condition	Value of A	Result
A IS NULL	10	FALSE
A IS NOT NULL	10	TRUE
A IS NULL	NULL	TRUE
A IS NOT NULL	NULL	FALSE

17.1.5.4.5.4. Null values in conditional expressions

Null values in conditional expressions are the NULL values in `= NULL`, `! = NULL`, `NULL =`, and `NULL ! =` conditions. These `NULL` values are used for logical evaluation. If conditions evaluate to `UNKNOWN`, no rows are returned.

In ApsaraDB for OceanBase, use the `IS NULL` comparison operator to test for null values. This operator returns `TRUE` or `FALSE`. However, `UNKNOWN` that is returned for null values in conditional expressions is different from `FALSE`. `NOT FALSE` evaluates to `TRUE`, but `NOT UNKNOWN` still evaluates to `UNKNOWN`.

The following table lists the results that are returned for conditional expressions based on the A value.

Condition	A value	Result
A = NULL	10	UNKNOWN
A ! = NULL	10	UNKNOWN
A = NULL	NULL	UNKNOWN
A ! = NULL	NULL	UNKNOWN
A = 10	NULL	UNKNOWN
A ! = 10	NULL	UNKNOWN

No rows are returned if a condition that evaluates to `UNKNOWN` is used in the `WHERE` clause of a `SELECT` statement.

17.1.5.4.6. Comments

17.1.5.4.6.1. Overview

In ApsaraDB for OceanBase, you can create three types of comments:

- Comments in SQL statements: The comments of this type are stored as part of the application code that is used to execute the SQL statements.
- Comments on schema and non-schema objects: The comments of this type and the object metadata are stored in a data dictionary.
- Hints: Hints are comments that are located in SQL statements and pass instructions to the ApsaraDB for OceanBase optimizer.

17.1.5.4.6.2. Comments in SQL statements

Comments make applications easier to read and maintain. For example, you can create a comment in a statement to describe the purpose of the statement in an application. Only hints in SQL statements affect how the statements are executed. The other comments in SQL statements do not affect how the statements are affected.

You can add a comment between keywords, parameters, or punctuations in a statement. You can use two methods to add a comment:

- Add a comment that starts with a slash and an asterisk (/*). The text of the comment follows the slash and the asterisk (/*). The text can be distributed across multiple lines. The comment must end with an asterisk and a slash (*/). You do not need to use white-space characters or line feeds to separate the start and end symbols from the text.
- Add a comment that starts with two hyphens (--). The text of the comment follows the hyphens (--). The text can be distributed in only one line. The comment must end with a line feed.

An SQL statement can contain multiple comments of the preceding two styles. The comment text can contain all of the printable characters in your database character set.

The following example is used to demonstrate multiple forms of comments, each of which starts with a slash and an asterisk (/*):

```
SELECT last_name, employee_id, salary + NVL(commission_pct, 0),
            job_id, e.department_id
  /* Select all employees whose compensation is
  greater than that of Pataballa. */
  FROM employees e, departments d
  /*The DEPARTMENTS table is used to get the department name. */
  WHERE e.department_id = d.department_id
    AND salary + NVL(commission_pct,0) >   /* Subquery:       */
        (SELECT salary + NVL(commission_pct,0)
          /* total compensation is salary + commission_pct */
          FROM employees
          WHERE last_name = 'Pataballa')
  ORDER BY last_name, employee_id;
```

The statement in the following example contains multiple forms of comments, each of which starts with two hyphens (--):

```
SELECT last_name,                                  -- select the name
       employee_id                                 -- employee id
       salary + NVL(commission_pct, 0),            -- total compensation
       job_id,                                     -- job
       e.department_id                             -- and department
  FROM employees e,                                -- of all employees
       departments d
  WHERE e.department_id = d.department_id
    AND salary + NVL(commission_pct, 0) >          -- whose compensation
                                                   -- is greater than
        (SELECT salary + NVL(commission_pct,0)     -- the compensation
          FROM employees
          WHERE last_name = 'Pataballa')           -- of Pataballa
  ORDER BY last_name                               -- and order by last name
           employee_id                             -- and employee id.
;
```

17.1.5.4.6.3. Comments on schema objects and non-schema objects

You can execute the `COMMENT` statement to associate comments with schema objects (tables, views, materialized views, operators, and index types) or non-schema objects (editions). You can also create comments for columns of table schema objects. Comments associated with schema objects and non-schema objects are stored in your data dictionary.

```
Syntax:
COMMENT ON {TABLE table | COLUMN column | INDEXTYPE indextype
    | OPERATOR operator | VIEW view} IS string
```

For example, you want to associate a comment with the table named user:

```
COMMENT ON TABLE test.user is "This is a table that records user information";
```

For example, you want to create a comment for the user_name column:

```
COMMENT ON COLUMN test.user.user_name is "The user names are recorded";
```

17.1.5.4.6.4. Hint

Hint overview

Hints are comments that are located in SQL statements and pass instructions to the ApsaraDB for OceanBase optimizer or server. The optimizer or the server can generate specific execution plans based on hints. In general, the optimizer chooses the optimal execution plan for your query without the need to use hints. However, in some scenarios, the optimizer-generated execution plan may not meet your requirements. In these scenarios, you can use hints to specify the execution plans that are to be generated.

We recommend that you do not use hints if possible. Use hints only after you have collected statistics about the relevant tables and have executed the `EXPLAIN PLAN` statement to evaluate the optimizer-generated plan that is not affected by hints. Query performance improvements in subsequent versions and changes to database conditions may cause hints in your code to have a significant impact on performance.

Use hints

In a statement block, only one comment can contain hints and the comment must follow the `SELECT`, `UPDATE`, `INSERT`, `MERGE`, or `DELETE` keyword.

The following hint syntax applies to comments in statement blocks:

```
/*+[hint text]*/
```

In terms of syntax, hints are a special type of SQL comments. The difference between comments and hints is that their marks are different. If a plus sign (+) is appended to the left mark of a comment, the comment is interpreted as a hint. If the server cannot recognize hints in SQL statements, the optimizer ignores the specified hints and uses the logic that is generated by the default execution plan. In addition, take note of the following point: Hints affect only the logic of optimizer-generated plans and do not affect the semantics of SQL statements.

Take note of the following rules when you define hints:

- The plus sign (+) causes the database to interpret a comment as a list of hints. The plus sign (+) must immediately follow the left mark of the comment. No white-space character is allowed.
- The white-space character between the plus sign (+) and the hint text is optional. If a comment contains multiple hints, use at least one white-space character to separate hints.

- Hints that contain spelling or syntax errors are ignored. However, the database does not ignore the other hints that are correctly specified in the same comment.
- Hints that do not follow the `DELETE`, `INSERT`, `MERGE`, `SELECT`, or `UPDATE` keyword are invalid.
- In a combination of hints, the hints that conflict with each other are invalid. However, the database does not ignore the other hints in the same comment.
- Hints are invalid when the database environment uses PostgreSQL version 1 or SQL Server version 1. For example, when the database environment uses triggers in Forms version 3, hints are invalid.

Specify query blocks in hints

You can specify an optional query block name in multiple hints to specify the query block on which the hint takes effect. In an outer query, you can use this syntax to specify a hint that applies to an inline view.

The parameters of a query block use the `@queryblock` syntax. In this syntax, `queryblock` is the identifier of the query block that is specified in your query. The `queryblock` identifier can be customized or system-generated. When you directly specify a hint to be applied in a query block, `@queryblock` is ignored.

- You can obtain the system-generated identifier by executing the `EXPLAIN PLAN` statement for your query.

  You can obtain the names of pre-transformation query blocks by executing the `EXPLAIN PLAN` statement for your query that uses the `NO_QUERY_TRANSFORMATION` hint.
- You can use `QB_NAME` to specify a custom name.

Specify global hints

Many hints can be applied to specific tables or indexes and more globally to the tables in a view or to some columns that are part of indexes. These global hints are specified by using the `tablespec` and `indexspec` syntax elements.

`tablespec` uses the following syntax:

```
[ view.[ view. ]... ]table
```

You must specify the table that you want to access exactly as it appears in the statement. If the statement uses a table alias, use the alias instead of the table name in the hint. However, even if the schema name appears in the statement, do not include the schema name in the table name that is used in the hint.

> **Notice**
>
> If you use the `tablespec` clause to specify global hints, the global hints do not take effect on the queries that use ANSI joins. This is because the optimizer generates additional views in the parsing process. Instead, you can use `@queryblock` to specify the query block to which the hint applies.

`indexspec` uses the following syntax:

```
{ index
| ( [ table. ]column [ [ table. ]column ]...)
}
```

When `tablespec` is followed by `indexspec` in the description of a hint, commas (,) that separate table names and index names are allowed but not required. Commas (,) that separate multiple occurrences of `indexspec` are also allowed but not required.

Lists of hints

Hints related to access paths

INDEX Hint

The `INDEX` hint instructs the optimizer to scan a specified table based on the index. You can use the `INDEX` hint for function-based, domain, B-tree, bitmap, and bitmap join indexes.

The `INDEX` hint uses the following syntax:

```
/*+ INDEX ( [ @ queryblock ] tablespec [ indexspec [ indexspec ]... ] ) */
```

The behavior of the hint depends on the `indexspec` specification:

- If the `INDEX` hint specifies a single available index, the database scans based on this index. The optimizer does not consider a full table scan or a scan on another index on the table.
- If the `INDEX` hint specifies multiple available indexes, the optimizer compares the costs of scans base on each index in the list. Then, the optimizer performs an index scan at the lowest cost. If an access path that is generated based on multiple indexes cause the lowest cost among all table scan plans, this access path is used. The database does not consider a full table scan or a scan on an index that is not specified in the hint.
- If the `INDEX` hint specifies no indexes, the optimizer compares the costs of scans on each available index on the table. Then, the optimizer performs an index scan at the lowest cost. If an access path that is generated based on multiple indexes cause the lowest cost among all table scan plans, this access path is used. The optimizer does not consider a full table scan.

The following statement provides an example:

```
SELECT /*+ INDEX (employees emp_department_ix)*/ employee_id, department_id
  FROM employees
  WHERE department_id > 50;
```

FULL Hint

The `FULL` hint instructs the optimizer to perform a full table scan on the specified table.

The `FULL` hint uses the following syntax:

```
/*+ FULL ( [ @ queryblock ] tablespec ) */
```

The following statement provides an example:

```
SELECT /*+ FULL(e) */ employee_id, last_name
  FROM hr.employees e
  WHERE last_name LIKE :b1;
```

The database performs a full table scan on the **employees** table to execute this statement, even if an index on the **last_name** column is listed in the condition in the `WHERE` clause.

The **employees** table has an alias **e** in the `FROM` clause. Therefore, the hint must reference the table by using its alias rather than its name. Do not reference schema names in the hint even if the schema names are specified in the `FROM` clause.

Hints related to join orders

LEADING Hint

The `LEADING` hint instructs the optimizer to use the specified set of tables as the prefix in the execution plan. This hint can be used to specify the join order of tables. This hint is more versatile than the `ORDERED` hint.

The `LEADING` hint uses the following syntax:

```
/*+ LEADING ( [ @ queryblock ] tablespec [ tablespec ]...  ) */
```

The `LEADING` hint performs strict checks to ensure that tables are joined in the specified order. If the `table_name` that is specified in a `LEADING` hint does not exist, the hint is ignored. If duplicate tables are found in a `LEADING` hint, the hint is ignored. If the optimizer attempts to join tables but cannot locate a table to be joined, the join orders specified for this table and the tables after the table become invalid. The join orders specified for the tables before this table remain valid. If the specified table cannot be first joined in the specified order due to the dependencies in the join graph, the `LEADING` hint is ignored. If you specify two or more conflicting `LEADING` hints, these `LEADING` hints are ignored. If you specify an `ORDERED` hint, it overrides all `LEADING` hints.

The following statement provides an example:

```
SELECT /*+ LEADING(e j) */ *
    FROM employees e, departments d, job_history j
    WHERE e.department_id = d.department_id
      AND e.hire_date = j.start_date;
```

ORDERED Hint

The `ORDERED` hint instructs a database to join tables in the order in which they appear in the `FROM` clause. We recommend that you use the `LEADING` hint. It is more versatile than the `ORDERED` hint.

The `ORDERED` hint uses the following syntax:

```
/*+ ORDERED */
```

When you omit the `ORDERED` hint in an SQL statement that performs a join operation, the optimizer chooses the order in which the tables are joined. However, the optimizer does not know the number of rows to be selected from each table. In this case, you can use the `ORDERED` hint to specify a join order. This allows you to select internal and external tables in a better way than the optimizer. If you rewrite the specified `ORDERED` hint, the tables are joined in the order as you rewrite in the `FROM` clause of the statement.

Hints related to join operations

USE_MERGE Hint

The `USE_MERGE` hint instructs the optimizer to join each specified table with another row resource by using a `sort-merge` join. We recommend that you use the `USE_NL` and `USE_MERG` hints when you use the `LEADING` and `ORDERED` hints. The optimizer uses these hints if the referenced table is the internal table of a join operation. The hints are ignored if the referenced table is an external table.

The `USE_MERGE` hint uses the following syntax:

```
/*+ USE_MERGE ( [ @ queryblock ] tablespec [ tablespec ]... ) */
```

The `USE_MERGE` hint uses the MERGE-JOIN algorithm if a table is specified as an internal table. ApsaraDB for OceanBase allows you to use the MERGE-JOIN algorithm only if a `join condition` is used to specify the equivalence relation between two fields in the two tables. If you attempt to join two tables without such join conditions, the `USE_MERGE` hint is invalid.

The following statement provides an example of the `USE_MERGE` hint:

```
SELECT /*+ USE_MERGE(employees departments) */ *
  FROM employees, departments
  WHERE employees.department_id = departments.department_id;
```

NO_USE_MERGE Hint

The `NO_USE_MERGE` hint instructs the optimizer to exclude the joins specified in the `USE_MERGE` hint when the optimizer uses a specified table as an internal table and joins the specified table with another row resource.

The `NO_USE_MERGE` hint uses the following syntax:

```
/*+ NO_USE_MERGE ( [ @ queryblock ] tablespec [ tablespec ]... ) */
```

The following statement provides an example of the `NO_USE_MERGE` hint:

```
SELECT /*+ NO_USE_MERGE(e d) */ *
  FROM employees e, departments d
  WHERE e.department_id = d.department_id;
```

USE_HASH Hint

The `USE_HASH` hint instructs the optimizer to join each specified table with another row resource by using the HASH-JOIN algorithm.

The `USE_HASH` hint uses the following syntax:

```
/*+ USE_HASH ( [ @ queryblock ] tablespec [ tablespec ]... ) */
```

The following statement provides an example of the `USE_HASH` hint:

```
SELECT /*+ USE_HASH(l h) */ *
  FROM orders h, order_items l
  WHERE l.order_id = h.order_id
    AND l.order_id > 2400;
```

NO_USE_HASH Hint

The `NO_USE_HASH` hint instructs the optimizer to exclude the joins specified in the `USE_HASH` hint when the optimizer uses a specified table as an internal table and joins the specified table to another row resource.

The `NO_USE_HASH` hint uses the following syntax:

```
/*+ NO_USE_HASH ( [ @ queryblock ] tablespec [ tablespec ]... ) */
```

The following statement provides an example of the `NO_USE_HASH` hint:

```
SELECT /*+ NO_USE_HASH(e d) */ *
  FROM employees e, departments d
  WHERE e.department_id = d.department_id;
```

USE_NL Hint

The `USE_NL` hint instructs the optimizer to join each specified table to another row resource by using a nested loop join. This hint also instructs the optimizer to use the specified table as an internal table by using the NL-JOIN algorithm. We recommend that you use the `USE_NL` and `USE_MERG` hints together with the `LEADING` and `ORDERED` hints. The optimizer uses these hints when the referenced table is the internal table of a join. The hints are ignored if the referenced table is an external table.

The `USE_NL` hint uses the following syntax:

```
/*+ USE_NL ( [ @ queryblock ] tablespec [ tablespec ]... ) */
```

In the following example, a hint is used to forcibly execute a nested loop and access **orders** through a full table scan. The filter condition `l.order_id = h.order_id` is applied to every row. If a row meets the filter condition, **order_items** is queried through the index **order_id**:

```
SELECT /*+ USE_NL(l h) */ h.customer_id, l.unit_price * l.quantity
  FROM orders h, order_items l
  WHERE l.order_id = h.order_id;
```

NO_USE_NL Hint

The `NO_USE_NL` hint instructs the optimizer to exclude nested loop joins when the optimizer uses a specified table as an internal table and joins the specified table to another row resource.

The `NO_USE_NL` hint uses the following syntax:

```
/*+ NO_USE_NL ( [ @ queryblock ] tablespec [ tablespec ]... ) */
```

The following statement provides an example of the `NO_USE_NL` hint:

```
SELECT /*+ NO_USE_NL(e d) */ *
  FROM employees e, departments d
  WHERE e.department_id = d.department_id;
```

USE_BNL Hint

The `USE_BNL` hint instructs the optimizer to join each specified table to another row resource by using a block-nested loop join. The hint also instructs the optimizer to use the specified table as an internal table by using the BNL-JOIN algorithm. We recommend that you use the `USE_BN` hint together with the `LEADING` and `ORDERED` hints. The optimizer uses these hints when the referenced table is the internal table of a join. The hints are ignored if the referenced table is an external table.

The `USE_BNL` hint uses the following syntax:

```
/*+ USE_BNL ( [ @ queryblock ] tablespec [ tablespec ]... ) */
```

In the following example, a hint is used to forcibly execute a block nested loop and access **orders** through a full table scan. The filter condition `l.order_id = h.order_id` is applied to every row. If a row meets the filter condition, **order_items** is queried through the index **order_id**:

```
SELECT /*+ USE_BNL(l h) */ h.customer_id, l.unit_price * l.quantity
  FROM orders h, order_items l
  WHERE l.order_id = h.order_id;
```

NO_USE_BNL Hint

The `NO_USE_BNL` hint instructs the optimizer to exclude the joins specified in the `USE_BNL` hint when the optimizer uses a specified table as an internal table and joins the specified table to another row resource.

The `NO_USE_BNL` hint uses the following syntax:

```
/*+ NO_USE_BNL ( [ @ queryblock ] tablespec [ tablespec ]... ) */
```

The following statement provides an example of the `NO_USE_BNL` hint:

```
SELECT /*+ NO_USE_BNL(e d) */ *
  FROM employees e, departments d
  WHERE e.department_id = d.department_id;
```

Hints related to parallel execution

PARALLEL Hint

The `PARALLEL` hint is a statement-level hint that instructs the optimizer to specify the number of parallel servers that can be used for parallel operations. The PARALLEL hint overwrites the value of the `PARALLEL_DEGREE_ POLICY` initialization parameter. The PARALLEL hint is applicable to the `SELECT`, `INSERT`, `MERGE`, `UPDATE`, and `DELETE` portions of a statement and to the table scan portion. If a parallel constraint is violated, the `PARALLEL` hint is ignored.

The `PARALLEL` hint uses the following syntax:

```
/*+ PARALLEL [ ( DEFAULT | AUTO | MANUAL | integer ) ] */
```

> **Notice**
>
> If sorting or grouping is also performed, the number of servers that can be used is twice the value specified in the `PARALLEL` hint.

You can set the following parameters in the `PARALLEL` hint:

- `PARALLEL`: the degree of parallelism that is calculated by the database. The value can be 2 or a greater value. The statement is always executed as a set of operations that are performed in parallel.
- `PARALLEL(DEFAULT)`: the degree of parallelism that is calculated by the optimizer. The parameter value is equal to the product of the value of the `PARALLEL_THREADS_PER_CPU` initialization parameter and the number of CPUs available on all of the participating instances.
- `PARALLEL(AUTO)`: the degree of parallelism that is calculated by the database. The value can be greater than or equal to 1. If the calculated degree of parallelism is 1, the statement is executed as a set of operations that are performed in sequence.
- `PARALLEL(MANUAL)`: the degree of parallelism that the optimizer is forced to use. The degree of parallelism is specified by the parallel settings of the objects in the statement.
- `PARALLEL(integer)`: the degree of parallelism that is used by the optimizer and specified by the `integer` parameter. The value of the integer parameter is an integer.

In the following example, the database calculates the degree of parallelism and the statement is always executed as a set of operations that are performed in parallel:

```
SELECT /*+ PARALLEL */ last_name
  FROM employees;
```

In the following example, the database calculates the degree of parallelism, but the degree of parallelism is 1. Therefore, the statement is executed as a set of operations that are performed in sequence:

```
SELECT /*+ PARALLEL (AUTO) */ last_name
  FROM employees;
```

In the following example, 5 is specified in the statement as the degree of parallelism and takes effect on the current table. The `PARALLEL` hint recommends that the optimizer use 5 as the degree of parallelism:

```
CREATE TABLE parallel_table (col1 number, col2 VARCHAR2(10)) PARALLEL 5;
SELECT /*+ PARALLEL (MANUAL) */ col2
 FROM parallel_table;
```

USE_PX Hint

The `USE_PX` hint instructs the server to execute SQL statements in PX mode. In PX mode, multithreading is allowed when statements are executed. In general, the `USE_PX` hint is used in conjunction with the `PARALLEL` hint.

The `USE_PX` hint uses the following syntax:

```
/*+ USE_PX  */
```

Example:

```
SELECT /*+ USE_PX PARALLEL(4)*/ e.department_id, sum(e.salary)
 FROM employees e
 WHERE e.department_id = 1001;
 GROUP BY e.department_id;
```

NO_USE_PX Hint

The `NO_USE_PX` hint instructs the server not to use the PX mode to execute SQL statements.

The `NO_USE_PX` hint uses the following syntax:

```
/*+ NO_USE_PX  */
```

Example:

```
SELECT /*+ NO_USE_PX*/ e.department_id, sum(e.salary)
 FROM employees e
 WHERE e.department_id = 1001;
 GROUP BY e.department_id;
```

PQ_DISTRIBUTE Hint

The `PQ_DISTRIBUTE` hint instructs the optimizer on how to distribute rows between producer (query) servers and consumer (load) servers. You can use this hint to control row distribution for joins or loads. The `PQ_DISTRIBUTE` hint uses the following syntax:

```
/*+ PQ_DISTRIBUTE
  ( [ @ queryblock ] tablespec
    { distribution | outer_distribution inner_distribution }
  ) */
```

Control load distribution

You can control the row distribution for parallel `INSERT ... SELECT` and `CREATE TABLE ... AS SELECT` statements to determine how rows are distributed between producer (query) servers and consumer (load) servers. Use the upper branch of the syntax to specify a distribution method. The following table lists the values and semantics of distribution methods.

Distribution method	Description
NONE	Row distribution is not performed. This means that query and load operations are combined on each query server. Each server loads all of the partitions. In this distribution method, row distribution is not performed. This helps you avoid the overhead of row distribution when no skew occurs. Skew may occur due to null fields. Skew may also occur because a predicate in the statement filters out all of the rows that are evaluated by the query. If skew occurs due to this distribution method, use `RANDOM` or `RANDOM_ LOCAL` distribution instead. **Notice** Use this distribution method with caution. A minimum of 512 KB Program Global Area (PGA) memory is required for the loading of each process. If compression is also used, each server consumes about 1.5 MB PGA memory.
PARTITION	This method uses the partition information of `tablespec` to distribute rows from query servers to consumer servers. Use this method when the action of combining query and load operations is not possible or expected. Use this method also when the number of partitions being loaded is greater than or equal to the number of load servers and no skew occurs. If no skew occurs, the input data is evenly distributed across the partitions being loaded.
RANDOM	This method distributes rows from producer servers to consumer servers in a round-robin manner. Use this method when the input data is highly skewed.
RANDOM_LOCAL	This method distributes rows from producer servers to a group of servers that are responsible for maintaining a given group of servers. Two or more servers can load the same partition, but none of the servers loads all of the partitions. Use this distribution method when the input data is skewed and query and load operations cannot be combined due to memory limits.

For example, in the following direct-load INSERT operation, the query and load portions of the operation are combined on each query server:

```
INSERT /*+ APPEND PARALLEL(target_table, 16) PQ_DISTRIBUTE(target_table, NONE) */
  INTO target_table
  SELECT * FROM source_table;
```

In the following example, the optimizer uses the partitioning of the **target_table** table to distribute rows when the table is created:

```
CREATE /*+ PQ_DISTRIBUTE(target_table, PARTITION) */ TABLE target_table
  NOLOGGING PARALLEL 16
  PARTITION BY HASH (l_orderkey) PARTITIONS 512
  AS SELECT * FROM source_table;
```

Control the distribution for joins

You can specify two methods to control the distribution for joins. In the lower branch of the syntax, one of the distribution methods is for the outer table and the other distribution method is for the inner table:

- `outside_distribution` specifies the distribution method for the outer table.
- `inner_distribution` specifies the distribution method for the inner table.

The values for distribution methods are `HASH`, `BROADCAST`, `PARTITION`, and `NONE`. Only the six combinations of distribution methods in the following table are valid.

Distribution method	Description
HASH, HASH	The rows of each table are mapped to consumer query servers by using a hash function on the join keys. After the mapping is completed, each query server performs a join operation on a pair of result partitions. We recommend that you use this distribution method when the sizes of tables are comparable and the join operation is implemented by using a hash join or a sort merge join.
BROADCAST, NONE	All of the rows in the outer table are broadcast to each producer query server. The rows in the inner table are randomly distributed across partitions. We recommend that you use this distribution method when the size of the outer table is extremely smaller than that of the inner table. In some scenarios, the product of the inner table size and the number of query servers is greater than the outer table size. In general, we also recommend that you use this distribution method in these scenarios.
NONE, BROADCAST	All of the rows in the inner table are broadcast to each consumer query server. The rows in the outer table are randomly distributed across partitions. We recommend that you use this distribution method when the size of the inner table is extremely smaller than that of the outer table. In some scenarios, the product of the inner table size and the number of query servers is smaller than the outer table size. In general, we also recommend that you use this distribution method in these scenarios.

Distribution method	Description
PARTITION, NONE	The rows in the outer table are mapped based on the partitioning of the inner table. The inner table must be partitioned based on the join key. We recommend that you use this distribution method when the number of outer table partitions is equal to or nearly equal to a multiple of the number of query servers. For example, the number of partitions is 14 and the number of query servers is 15. **Notice** If the inner table is not partitioned or an equijoin is not performed on the inner table based on the partition key, the optimizer ignores this hint.
NONE, PARTITION	The rows in the inner table are mapped based on the partitioning of the outer table. The outer table must be partitioned based on the join key. We recommend that you use this distribution method when the number of outer table partitions is equal to or nearly equal to a multiple of the number of query servers. For example, the number of partitions is 14 and the number of query servers is 15. **Notice** If the outer table is not partitioned or an equijoin is not performed on the outer table based on the partition key, the optimizer ignores this hint.
NONE, NONE	Each query server performs a join operation on a pair of matching partitions. Each table contains a partition that consists of the pair. The two tables must be equally joined based on the join keys.

In the following example, a hash join is implemented on the two given tables **r** and **s** and the query contains a hint that instructs the system to use hash distribution:

```
SELECT /*+ORDERED PQ_DISTRIBUTE(s HASH, HASH) USE_HASH (s)*/ column_list
  FROM r,s
  WHERE r.c=s.c;
```

To broadcast the outer table **r**, execute the following query statement:

```
SELECT /*+ORDERED PQ_DISTRIBUTE(s BROADCAST, NONE) USE_HASH (s) */ column_list
  FROM r,s
  WHERE r.c=s.c;
```

Hints related to query policies

USE_JIT Hint

The `USE_JIT` hint instructs the server to forcibly execute expressions by using the JIT compilation when the server is executing SQL statements.

The `USE_JIT` hint uses the following syntax:

```
/*+ USE_JIT  */
```

The following statement provides an example:

```
SELECT /*+ USE_JIT*/ e.department_id, sum(e.salary)
 FROM employees e
 WHERE e.department_id = 1001;
 GROUP BY e.department_id;
```

NO_USE_JIT Hint

The `NO_USE_JIT` hint prevents the server from using the JIT compilation to execute expressions when the server is executing SQL statements.

The `NO_USE_JIT` hint uses the following syntax:

```
/*+ NO_USE_JIT*/
```

The following statement provides an example:

```
SELECT /*+NO_USE_JIT*/ e.department_id, sum(e.salary)
 FROM employees e
 WHERE e.department_id = 1001;
 GROUP BY e.department_id;
```

USE_HASH_AGGREGATION Hint

The `USE_HASH_AGGREGATION` hint instructs the optimizer to forcibly execute the SQL statement by using the hash aggregation algorithm when the optimizer is generating a plan.

The `USE_HASH_AGGREGATION` hint uses the following syntax:

```
/*+ USE_HASH_AGGREGATION */
```

The following statement provides an example:

```
SELECT /*+ USE_HASH_AGGREGATION */ e.department_id, sum(e.salary)
 FROM employees e
 WHERE e.department_id = 1001;
 GROUP BY e.department_id;
```

NO_USE_HASH_AGGREGATION Hint

The `NO_USE_HASH_AGGREGATION` hint prevents the optimizer from using the hash aggregation algorithm to run an SQL statement when the optimizer is executing the statement.

The `NO_USE_HASH_AGGREGATION` hint uses the following syntax:

```
/*+ NO_USE_HASH_AGGREGATION */
```

The following statement provides an example:

```
SELECT /*+ NO_USE_HASH_AGGREGATION */ e.department_id, sum(e.salary)
 FROM employees e
 WHERE e.department_id = 1001;
 GROUP BY e.department_id;
```

USE_LATE_MATERIALIZATION Hint

The `USE_LATE_MATERIALIZATION` hint instructs the optimizer to delay view materialization.

The `USE_LATE_MATERIALIZATION` hint uses the following syntax:

```
/*+ USE_LATE_MATERIALIZATION  */
```

The following statement provides an example:

```
SELECT /*+ USE_LATE_MATERIALIZATION*/ e.department_id, sum(e.salary)
 FROM employees e
 WHERE e.department_id = 1001;
 GROUP BY e.department_id;
```

NO_USE_LATE_MATERIALIZATION Hint

The `NO_USE_LATE_MATERIALIZATION` hint instructs the optimizer to disable the delay of view materialization.

The `NO_USE_LATE_MATERIALIZATION` hint uses the following syntax:

```
/*+ NO_USE_LATE_MATERIALIZATION  */
```

The following statement provides an example:

```
SELECT /*+ NO_USE_LATE_MATERIALIZATION*/ e.department_id, sum(e.salary)
 FROM employees e
 WHERE e.department_id = 1001;
 GROUP BY e.department_id;
```

USE_NL_MATERIALIZATION Hint

The `USE_NL_MATERIALIZATION` hint forcibly instructs the optimizer to generate a materialize operator to cache data when the optimizer specifies a table as an internal table or a subtree.

The `USE_NL_MATERIALIZATION` hint uses the following syntax:

```
/*+ USE_NL_MATERIALIZATION ( [ @ queryblock ] tablespec [ tablespec ]... ) */
```

The following statement provides an example:

```
SELECT /*+ USE_NL_MATERIALIZATION(departments) */ *
  FROM employees, departments
  WHERE employees.department_id = departments.department_id;
```

NO_USE_NL_MATERIALIZATION Hint

The `NO_USE_NL_MATERIALIZATION` hint prevents the optimizer from generating a materialize operator to cache data when the optimizer specifies a table as an internal table or a subtree.

The `NO_USE_NL_MATERIALIZATION` hint uses the following syntax:

```
/*+ NO_USE_NL_MATERIALIZATION ( [ @ queryblock ] tablespec [ tablespec ]... ) */
```

The following statement provides an example:

```
SELECT /*+ NO_USE_NL_MATERIALIZATION(departments) */ *
  FROM employees, departments
  WHERE employees.department_id = departments.department_id;
```

Hints related to query transformation

NO_REWRITE Hint

The `NO_REWRITE` hint instructs the optimizer to disable query rewrites for the query block and override the setting of the `QUERY_REWRITE_ENABLED` parameter.

The `NO_REWRITE` hint uses the following syntax:

```
/*+ NO_REWRITE [ ( @ queryblock ) ] */
```

The following statement provides an example:

```
SELECT /*+ NO_REWRITE */ sum(s.amount_sold) AS dollars
  FROM sales s, times t
  WHERE s.time_id = t.time_id
  GROUP BY t.calendar_month_desc;
```

NO_EXPAND Hint

The `NO_EXPAND` hint prevents the optimizer from considering `OR` expansion for queries that have `OR` conditions or `IN` lists in the `WHERE` clause. Usually, the optimizer uses `OR` expansion when the optimizer decides that the cost of using `OR` extension is lower than not using it.

The `NO_EXPAND` hint uses the following syntax:

```
/*+ NO_EXPAND [ ( @ queryblock ) ] */
```

The following statement provides an example:

```
SELECT /*+ NO_EXPAND */ *
  FROM employees e, departments d
  WHERE e.manager_id = 108
     OR d.department_id = 110;
```

USE_CONCAT Hint

The `USE_CONCAT` hint instructs the optimizer to transform combined `OR` conditions in the `WHERE` clause of a query into a compound query by using the `UNION ALL` operator. If this hint is not used, this transformation occurs only if the cost of the query based on concatenations is lower than that without concatenations. The `USE_CONCAT` hint overrides the cost consideration.

The `USE_CONCAT` hint uses the following syntax:

```
/*+ USE_CONCAT [ ( @ queryblock ) ] */
```

The following statement provides an example:

```
SELECT /*+ USE_CONCAT */ *
  FROM employees e
  WHERE manager_id = 108
     OR department_id = 110;
```

MERGE Hint

The `MERGE` hint allows you to merge views in a query.

The `MERGE` hint uses the following syntax:

```
/*+ MERGE [ ( @ queryblock )  | ( [ @ queryblock ] tablespec ) ] */
```

If the query block of a view includes the `GROUP BY` clause or `DISTINCT` operator in the `SELECT` list, the optimizer can merge the view into the accessing statement only if complex view merging is enabled. If an `IN` subquery is uncorrelated, you can also use complex merging to merge the subquery into the accessing statement. The following statement provides an example:

```
SELECT /*+ MERGE(v) */ e1.last_name, e1.salary, v.avg_salary
   FROM employees e1,
        (SELECT department_id, avg(salary) avg_salary
           FROM employees e2
           GROUP BY department_id) v
   WHERE e1.department_id = v.department_id
     AND e1.salary > v.avg_salary
   ORDER BY e1.last_name;
```

When you use the `MERGE` hint without an argument, you must place the hint in the query block of a view. When you use

the `MERGE` hint with the view name as an argument, you must place the hint in the surrounding query.

NO_MERGE Hint

The `NO_MERGE` hint instructs the optimizer not to combine the outer query and inline view queries into a single query.

The `NO_MERGE` hint uses the following syntax:

```
/*+ NO_MERGE [ ( @ queryblock )  | ( [ @ queryblock ] tablespec ) ] */
```

This hint affects how you access the view. For example, if you execute the following statement, the view **seattle_dept** cannot be merged:

```
SELECT /*+ NO_MERGE(seattle_dept) */ e1.last_name, seattle_dept.department_name
  FROM employees e1,
       (SELECT location_id, department_id, department_name
          FROM departments
          WHERE location_id = 1700) seattle_dept
  WHERE e1.department_id = seattle_dept.department_id;
```

When you use the `NO_MERGE` hint in the view query block, you do not need to specify arguments for the hint.

When you specify the `NO_MERGE` hint in the surrounding query, you must specify the hint with the view name as an argument.

UNNEST Hint

The `UNNEST` hint instructs the optimizer not to nest but to merge the body of the subquery into the body of the query block that contains the hint. This allows the optimizer to consider the subquery and the hint together when the optimizer evaluates access paths and joins.

The `UNNEST` hint uses the following syntax:

```
/*+ UNNEST [ ( @ queryblock ) ] */
```

Before a subquery is unnested, the optimizer first verifies whether the subquery is valid. The subquery must pass heuristic and query optimization tests. When you use the `UNNEST` hint, the optimizer verifies the validity of only the subquery block. If the subquery block is valid, subquery unnesting is enabled even if heuristic and query optimization tests are not passed.

The following statement provides an example:

```
SELECT AVG(t1.c) FROM t1
WHERE t1.b >=
    (SELECT /*+unnest*/AVG(t2.b)
     FROM t2
     WHERE t1.a = t2.a);
```

NO_UNNEST Hint

The `NO_UNNEST` hint is used to disable subquery unnesting.

The `NO_UNNEST` hint uses the following syntax:

```
/*+ NO_UNNEST [ ( @ queryblock ) ] */
```

The following statement provides an example:

```
SELECT /*+no_unnest(@qb1)*/AVG(t1.c)
FROM t1 WHERE t1.b >=
    (SELECT /*+qb_name(qb1)*/AVG(t2.b)
     FROM t2)
     WHERE t1.a = t2.a);
```

PLACE_GROUP_BY Hint

The `PLACE_GROUP_BY` hint instructs the optimizer to use sequence replacement rules that are specified in the GROUP BY clause. In this case, the optimizer does not consider the cost increase caused by the transformation.

The `PLACE_GROUP_BY` hint uses the following syntax:

```
/*+ PLACE_GROUP_BY [ ( @ queryblock ) ] */
```

The following statement provides an example:

```
SELECT /*+place_group_by*/SUM(t1.c),SUM(t2.c) FROM t1, t2
WHERE t1.a = t2.a AND t1.b > 10 AND t2.b > 10
GROUP BY  t1.a;
```

NO_PLACE_GROUP_BY Hint

The `NO_PLACE_GROUP_BY` hint is used to disable the sequence transformation that is specified in the GROUP BY clause.

The `NO_PLACE_GROUP_BY` hint uses the following syntax:

```
/*+ NO_PLACE_GROUP_BY [ ( @ queryblock ) ] */
```

The following statement provides an example:

```
SELECT /*+no_place_group_by*/SUM(t1.c),SUM(t2.c) FROM t1, t2
WHERE t1.a = t2.a AND t1.b > 10 AND t2.b > 10
GROUP BY  t1.a;
```

NO_PRED_DEDUCE Hint

The `NO_PRED_DEDUCE` hint instructs the optimizer not to use predicates to deduce the transformation rules.

The `NO_PRED_DEDUCE` hint uses the following syntax:

```
/*+ NO_PRED_DEDUCE [ ( @ queryblock ) ] */
```

The following statement provides an example:

```
SELECT /*+no_pred_deduce(@qb1)*/ *
FROM  (
  SELECT /*+no_merge qb_name(qb1)*/ t1.a, t2.b
  FROM t1, t2
  WHERE t1.a = t2.a) v, t3
WHERE t3.a = 1 AND t3.a = v.a;
```

Other hints

QB_NAME Hint

Use the `QB_NAME` hint to specify a name for a query block. Then, you can use this name in the hint of an outer query or in the hint of an inline view to affect how queries are run on the tables that appear in the named query block. For more information about query block names, see the section "Specify query blocks in hints" in Hint overview.

The `QB_NAME` hint uses the following syntax:

```
/*+ QB_NAME ( queryblock ) */
```

Assume that two or more query blocks have the same name or different names are specified by two hints for the same query block. In this case, the optimizer ignores all of the names and hints that reference the query blocks. The system generates unique names for the query blocks that are not named by using the `QB_NAME` hint. These names can be displayed in plan tables and can also be used in the other query block hints.

In the following example, the `QB_NAME` hint is used:

```
SELECT /*+ QB_NAME(qb) FULL(@qb e) */ employee_id, last_name
  FROM employees e
  WHERE last_name = 'Smith';
```

READ_CONSISTENCY Hint

The `READ_CONSISTENCY` hint instructs a server to specify the mode of table reading for an SQL statement as weak consistency (specified by the

`WEAK` parameter) or strong consistency (specified by the `STRONG` parameter).

The `READ_CONSISTENCY` hint uses the following syntax:

```
/*+ READ_CONSISTENCY(WEAK[STRONG]) */
```

Example:

```
SELECT /*+ READ_CONSISTENCY(WEAK) */ *
  FROM employees
  WHERE employees.department_id = 1001;
```

FROZEN_VERSION Hint

The `FROZEN_VERSION` hint instructs a server to read the baseline data of a specified version.

The `FROZEN_VERSION` hint uses the following syntax:

```
/*+ FROZEN_VERSION (intnum) */
```

Example:

```
SELECT /*+ FROZEN_VERSION(1000) */ *
  FROM employees e
  WHERE e.department_id = 1001;
```

QUERY_TIMEOUT Hint

The `QUERY_TIMEOUT` hint instructs a server to specify the time-out period of executing an SQL statement. The unit for the time-out period is microseconds.

The `QUERY_TIMEOUT` hint uses the following syntax:

```
/*+ QUERY_TIMEOUT (intnum) */
```

In the following example, a time-out error is returned when the statement in the query fails to be fully executed within a second:

```
SELECT /*+ QUERY_TIMEOUT(1000000) */ *
  FROM employees e
  WHERE e.department_id = 1001;
```

LOG_LEVEL Hint

The `LOG_LEVEL` hint instructs a server to use a specified log level when an SQL statement is executed.

The `LOG_LEVEL` hint uses the following syntax:

```
/*+ LOG_LEVEL (']log_level[']) */
```

In the following example, the `DEBUG` log level is used for executing the SQL statement:

```
SELECT /*+ LOG_LEVEL(DEBUG) */ *
  FROM employees e
  WHERE e.department_id = 1001;
```

USE_PLAN_CACHE Hint

The `USE_PLAN_CACHE` hint specifies whether to use the plan cache mechanism when a server executes an SQL statement. The `NONE` parameter specifies that the plan cache mechanism is not used. The `DEFAULT` parameter specifies that the server settings determine whether to use the plan cache mechanism.

The `USE_PLAN_CACHE` hint uses the following syntax:

```
/*+ USE_PLAN_CACHE (NONE[DEFAULT]) */
```

In the following example, the plan cache mechanism is not implemented for the statement:

```
SELECT /*+ USE_PLAN_CACHE(NONE) */ *
  FROM employees e
  WHERE e.department_id = 1001;
```

TRANS_PARAM Hint

The `TRANS_PARAM` hint specifies whether to use the parameters that are specified by the `param` parameter when a server performs a transaction. Only the `FORCE_EARLY_LOCK_FREE` parameter at the transaction level is supported. The FORCE_EARLY_LOCK_FREE parameter specifies the setting of releasing row locks in advance. When `FORCE_EARLY_LOCK_FREE` is set to **TRUE**, the parameters that are specified by param are used. When FORCE_EARLY_LOCK_FREE is set to **FALSE**, the parameters that are specified by param are not used. Take note of the following point: The parameter names must be enclosed in single quotation marks ('). The parameter values must be also enclosed in single quotation marks ('), except when the parameter values are numeric values.

The `TRANS_PARAM` hint uses the following syntax:

```
/*+ TRANS_PARAM ['param' , 'param_value'] */
```

Example:

```
SELECT /*+ TRANS_PARAM('FORCE_EARLY_LOCK_FREE' 'TRUE') */ *
  FROM employees e
  WHERE e.department_id = 1001;
```

TRACING Hint

The `TRACING` hint instructs a server to implement `TRACING` for some operators in execution plans.

The `TRACING` hint uses the following syntax:

```
/*+ TRACING(TRACING_NUM_LIST)*/
```

Example:

```
SELECT /*+ TRACING(1) */ *
  FROM employees e
  WHERE e.department_id = 1001;
```

STAT Hint

The `STAT` hint specifies that `STAT` is used to display the information about some operators in execution plans.

The `STAT` hint uses the following syntax:

```
/*+ STAT(TRACING_NUM_LIST) */
```

Example:

```
SELECT /*+ STAT(1) */ *
  FROM employees e
  WHERE e.department_id = 1001;
```

TOPK Hint

The `TOPK` hint instructs a server to set the precision and the minimum number of rows for a fuzzy match. The value of the `PRECISION` parameter is an integer that ranges from 0 to 100. This parameter specifies the percentage of rows that are returned for a fuzzy match. The `MINIMUM_ROWS` parameter specifies the minimum number of rows that are returned.

The `TOPK` hint uses the following syntax:

```
/*+ TOPK(PRECISION MINIMUM_ROWS) */
```

Example:

```
SELECT /*+ TOPK(1,10) */ *
  FROM employees e
  WHERE e.department_id = 1001;
```

TRACE_LOG Hint

The `TRACE_LOG` hint instructs a server to collect trace logs. The trace logs are displayed when you run the `SHOW TRACE` command.

The `TRACE_LOG` hint uses the following syntax:

```
/*+ TRACE_LOG */
```

Example:

```
SELECT /*+ TRACE_LOG */ *
  FROM employees e
  WHERE e.department_id = 1001;
```

17.1.5.4.7. Database objects

17.1.5.4.7.1. Schema objects

A schema is a collection of data logical structures and schema objects. Each user of ApsaraDB for OceanBase has a schema and the schema has the same name as the user.

You can use SQL to create and process your database objects. Schema objects are divided into the following types:

- Constraints
- Database links
- Database triggers
- Indexes

- Object tables
- Object types
- Object views
- Packages
- Sequences
- Stored functions
- Stored procedures
- Synonyms
- Tables
- Views

17.1.5.4.8. Database naming conventions

17.1.5.4.8.1. Overview of the naming conventions of database objects

Some database objects consist of parts that you can or must name. For example, the columns in a table or view, indexes, table partitions and subpartitions, table integrity constraints, and objects that are stored within packages, including procedures and stored functions. This topic discusses the following content:

• Naming rules of database objects

• Examples of naming schema objects

• Naming guidelines of schema objects

17.1.5.4.8.2. Naming rules of database objects

Database object identifiers

Every database object has a name. In an SQL statement, you can use a quoted identifier or a nonquoted identifier to name a database object.

- A quoted identifier starts and ends with double quotation marks (`"`). If you name a schema object by using a quoted identifier, you must use double quotation marks (") when you reference this object.
- A nonquoted identifier does not contain punctuation marks.

However, database names, global database names, database link names, disk group names, and pluggable database (PDB) names are not case-sensitive and are stored in uppercase. If you specify such names by using quoted identifiers, the quotation marks are ignored.

> **Notice**
>
> We do not recommend that you use quoted identifiers to name database objects in ApsaraDB for OceanBase. These quoted identifiers are compatible with SQL Plus. However, other tools that manage database objects may fail to recognize these quoted identifiers.

Identifier usage rules

The following rules apply to both quoted and nonquoted identifiers unless otherwise specified:

Identifier length

An identifier is 1 to 128 bytes in length.

If an identifier contains multiple parts separated by periods (.), each part can be up to 128 bytes in length. Each period delimiter consumes one byte. Each double quotation mark consumes 1 byte. The following string provides an example:

```
"schema"."table"."column"
```

The string contains three parts: **schema**, **table**, and **column**. Each part can be up to 128 bytes in length. Each of the quotation marks and periods is a single-byte character. Therefore, the total length of the identifier in this example can be up to 392 bytes in length.

Use reserved words as identifiers

Nonquoted identifiers cannot be SQL reserved words in ApsaraDB for OceanBase. Quoted identifiers can be reserved words. However, we recommend that you do not use reserved words as quoted identifiers. Based on the tools that you plan to use to access database objects, the names may be further limited by other product-specific reserved words.

> **Notice**
>
> The reserved word `ROWID` is an exception to this rule. You cannot use the uppercase word `ROWID` as a column name, regardless of whether the word is quoted or nonquoted. You cannot use an all-caps quoted identifier as a column name. However, you can use a quoted identifier that contains one or more lowercase letters as a column name, for example, "Rowid" or "Rowid".

Use words with special meanings as identifiers

In ApsaraDB for OceanBase, the SQL language contains other words that have special meanings. These words include data types, schema names, function names, the virtual system table `DUAL`, and keywords. The keywords are the all-caps words in SQL statements, such as `DIMENSION`, `SEGMENT`, `ALLOCATE`, and `DISABLE`. These words are not reserved. However, ApsaraDB for OceanBase uses them inside the system in specific ways. Therefore, if you use these words as the names of objects and object parts, your SQL statements may be difficult to read and may lead to unpredictable results. In particular, do not use the names of SQL built-in functions as the names of schema objects and user-defined functions.

Use ASCII characters as identifiers

Characters in the ASCII character set provide optimal compatibility across different platforms and operating systems. Therefore, we recommend that you use these characters in database names, global database names, and database link names. You can use only the characters from the ASCII character set in the names of common users and common roles in a multitenant container database (CDB).

Characters in passwords

Passwords can contain multibyte characters, such as Chinese characters or Chinese punctuation marks.

Beginning of identifiers

Nonquoted identifiers must start with an alphabetic character from the database character set. Quoted identifiers can start with all characters.

Signs in identifiers

Nonquoted identifiers can contain only alphanumeric characters from the database character set and underscores (_). However, nonquoted identifiers that are used in database link names can contain periods (.) and at signs (@). Quoted identifiers can contain all characters, punctuation marks, and spaces. However, both quoted and nonquoted identifiers cannot contain double quotation marks (") or the null character (\0).

Limits of object names in namespaces

Tables, views, and private synonyms in the same namespace cannot have the same name.

Identifier case sensitivity

Nonquoted identifiers are not case-sensitive. ApsaraDB for OceanBase stores them in uppercase. Quoted identifiers are case-sensitive. By enclosing names in double quotation marks, you can assign the following names to different objects in the same namespace:

```
"employees"
"Employees"
"EMPLOYEES"
```

Nonquoted identifiers are not case-sensitive. Therefore, ApsaraDB for OceanBase considers the following names the same. You cannot use these names for different objects in the same namespace:

```
employees
EMPLOYEES
"EMPLOYEES"
```

Uppercase identifiers

When you store or compare all-caps identifiers, the uppercase form of each character in the identifiers is determined by applying the capitalization rules of the database character set. The language rules that are specified in the session setting `NLS_SORT` are not considered. The SQL function `UPPER`, rather than the function `NLS_UPPER`, is applied to the identifiers.

If you apply the capitalization rules of the database character set to natural languages, the capitalization result may be incorrect. For example, based on the capitalization rules of the database character set, the German lowercase letter `ß` does not have an uppercase form. Therefore, this letter does not change when you convert an identifier that contains this letter to an all-caps word.

The capitalization rules of the database character set ensure that identifiers are displayed in the same form regardless of the session language. To display an identifier correctly in a specific natural language, quote the identifier to keep the lowercase form or convert the letters to the correct uppercase letters in this language.

Column names

Columns in the same table or view cannot have the same name. However, columns in different tables or views can have the same name.

Procedure and function names

If the parameters in two procedures or functions in the same package have different quantities and data types, the procedures or functions can have the same name. Creating multiple procedures or functions with the same name using different parameters in the same package is called overloaded procedures or functions.

17.1.5.4.8.3. Examples of schema object names

The following examples provide some valid schema object names:

- `last_name`
- `horse`
- `hr.hire_date`
- `"EVEN THIS & THAT!"`
- `a_very_long_and_valid_name`

All the examples follow the rule illustrated in Rules for naming database objects.

17.1.5.4.8.4. Rules for naming schema objects

The following rules are useful for naming objects and their parts:

- Use full, descriptive, and pronounceable names or well-known abbreviations.
- Use the same naming rule for all objects.
- Assign the same name to the same object or attribute in different tables.

Make sure that you assign names that are both simple and descriptive to objects. If these objectives cannot be balanced, ensure that the name is descriptive. This is because database objects may be used for a long time. Assume that a table column is named **pmdd**. Users that use the database ten years later may not understand what information this column stores. However, the table column named **payment_due_date** is clearer to the users.

The object names based on the same naming rule help users understand the role that each table plays in your application. For example, assign names that start with **fin_** to all tables that belong to the **FINANCE** application.

Assign the same name to the same object in different tables. For example, if both the employee table and the department table have a column that stores the same department IDs, name the column **department_id** in both tables.

17.1.5.4.9. Methods of referencing database objects

17.1.5.4.9.1. Overview of database object references

This topic discusses how to reference schema objects and parts of them in the context of SQL statements. This topic provides:

- Reference schema objects
- Reference objects in a remote database
- Reference partition tables and indexes
- Reference object type attributes and methods
- Create remote database links

17.1.5.4.9.2. Reference schema objects

When an object is referenced in an SQL statement, ApsaraDB for OceanBase considers the context of the SQL statement and locates the object in the appropriate namespace. After the object is located, ApsaraDB for OceanBase executes the SQL statement to perform the specified operation on the object. If the named object cannot be located in the appropriate namespace, an error is returned.

Reference objects in users' schemas

The following example is used to illustrate how ApsaraDB for OceanBase resolves object references in SQL statements:

Execute the following statement to add a row of data to the table named **departments**:

```
INSERT INTO departments
VALUES (280, 'ENTERTAINMENT_CLERK', 206, 1700);
```

Based on the context of the SQL statement, **departments** may represent one of the following items in the database:

- A table in your schema

- A view in your schema
- A private synonym of a table or a view
- A public synonym

ApsaraDB for OceanBase always attempts to resolve the referenced objects in the namespaces of your schema before it considers the namespaces outside your schema. In this example, ApsaraDB for OceanBase attempts to resolve the object:

1. Firstly, ApsaraDB for OceanBase attempts to locate the object in the namespace of your schema that contains tables, views, and private synonyms. If the object is a private synonym, ApsaraDB for OceanBase locates the object that the synonym represents. The object can be in your own schema, another schema, or another database. The object can also be another synonym. In this case, ApsaraDB for OceanBase locates the object that the synonym represents.
2. If the object is located in the namespace, ApsaraDB for OceanBase attempts to execute the SQL statement on the object. In this example, ApsaraDB for OceanBase attempts to add a row of data to **departments**. If the type of the object is not the required type of the SQL statement, ApsaraDB for OceanBase returns an error. In this example, **departments** must be a table, view, or private synonym that can be resolved as a table or a view. If **departments** is a sequence, ApsaraDB for OceanBase returns an error.
3. If the object is not found in all of the namespaces that have been searched so far, ApsaraDB for OceanBase searches the namespace that contains public synonyms. If the object is in the namespace that contains public synonyms, ApsaraDB for OceanBase attempts to execute the statement on the object. If the type of the object is not the required type of the SQL statement, ApsaraDB for OceanBase returns an error. In this example, if **departments** is a public synonym that represents a sequence, ApsaraDB for OceanBase returns an error.

Assume that the public synonym has dependent tables or user-defined types. In this case, you cannot create an object with the same name as the synonym in the same schema as that of the dependent object.

On the contrary, assume that the synonym does not have dependent tables or user-defined types. In this case, you can create an object with the same name as the synonym in the same schema as that of the dependent object. ApsaraDB for OceanBase invalidates all of the dependent objects, and revalidates them when they are accessed the next time.

Reference objects in other schemas

To reference objects in schemas other than your schema, add the schema name to the beginning of the object name:

```
schema.object
```

In the following example, the **employees** table is deleted from the schema named **hr**:

```
DROP TABLE hr.employees;
```

17.1.5.4.9.3. Reference objects in a remote database

Create database links

To reference an object in a database other than the local database, specify the name of the database link that points to the desired database after the object name. A database link is a schema object that enables ApsaraDB for OceanBase to connect to a remote database and access the objects in the remote database.

You can use the CREATE DATABASE LINK statement to create a database link. When you use this statement, you must specify the following database link information:

- The tenant name, username, and password.
- The network address and port number.

ApsaraDB for OceanBase stores the preceding information in a data dictionary.

Limits on creating links

1. You can only create database links to connect an Oracle tenant to another Oracle tenant in ApsaraDB for OceanBase. Currently, ApsaraDB for OceanBase does not allow you to create a link that connect an ApsaraDB for OceanBase database to an external Oracle database.
2. A tenant can create and delete database links for multiple times. However, a maximum of 15 database links are allowed under a tenant at the same time.

Format of database connection strings

```
user_name@tenant_name IDENTIFIED BY password HOST 'ip:port'
```

- **user_name**: The username.
- **tenant_name**: The tenant name.
- **password**: The password.
- **ip**: The network address.
- **port**: The port number.

For example, you can execute the following statement to create a database link:

```
CREATE DATABASE LINK my_link CONNECT TO
root@oracle  IDENTIFIED BY abcdef HOST '192.168.0.0:1521';
```

Reference database links

When you issue an SQL statement that contains a database link, you need to use the complete database link name as stored in the data dictionary. For more information about database link names, see Create database links.

When ApsaraDB for OceanBase connects to a remote database, ApsaraDB for OceanBase uses a database connection string to access the remote database. If the connection succeeds by using the database string, username, and password, ApsaraDB for OceanBase follows the rules in Reference schema objects to access the specified object in the remote database. However, take note of the following limits:

1. You can only execute read-only statements.
2. You can only access table objects but not other objects such as views and sequences.
3. When you access a table object, you must explicitly specify the database name, for example, `test.t1@my_link`.
4. The following plans and operators are not supported:
   - You cannot perform the `RESCAN` operation on the remote database.
   - You cannot execute the operators such as `NESTED LOOP JOIN`, `SEMI JOIN`, `ANTI JOIN`, and `SUBPLAN FILTER` on the remote database.
   - If the preceding problems occur, you can attempt to run the `EXPLAIN PLAN` statement to view the original plan and the SQL statements sent to the remote cluster for execution, and adjust the plan by using a hint.

17.1.5.4.9.4. Reference partitioned tables and indexes

You can partition tables and indexes. When you partition a table, these schema objects consist of a number of parts called partitions. All the partitions have the same logical attributes. For example, all partitions in a table share the same column and constraint definitions, whereas all partitions in an index share the same index columns.

Partition-extended and subpartition-extended names allow you to perform some partition-level and subpartition-level operations. For example, you can delete all rows from only one partition or subpartition. To manage partitions or subpartitions that do not have extended names, specify a range by using the `WHERE` clause. For range-partitioned and list-partitioned tables, you may encounter difficulties when you attempt to use conditional statements to phrase a partition-level operation. The difficulty increases if multiple columns are used as the range partition keys. Hash partitions and subpartitions are based on system-defined hash functions. Therefore, you may encounter more difficulties in using conditional statements on these partitions and subpartitions.

Partition-extended names allow you to use partitions in the same way as you use tables. This method allows you to build partition-level access control mechanisms by granting (or revoking) permissions on these views to (or from) other users or roles. This feature is useful when you query data in range-partitioned tables. To use a partition as a table, create a view by selecting data from the partition, and then use the view as a table.

Syntax

If the `partition_extended_name` or `subpartition_extended_name` element appears in the syntax of an SQL statement, you can use these elements to specify partition-extended or subpartition-extended table names.
Syntax of `partition_extended_name`:

```
PARTITION partition
|
PARTITION FOR ( partition_key_value [, partition_key_value]... )
```

Syntax of `subpartition_extended_name`:

```
SUBPARTITION subpartition
|
SUBPARTITION FOR ( subpartition_key_value [, subpartition_key_value]... )
```

Partition or subpartition names in data manipulation language (DML) statements, such as `INSERT`, `UPDATE`, `DELETE`, and `ANALYZE`, must be included in parentheses (). This small distinction is reflected in the `partition_extension_clause` element: In `partition_extended_name`, `subpartition_extended_name`, and `partition_extension_clause`, the `PARTITION FOR` and `SUBPARTITION FOR` clauses allow you to reference partitions without using their names. They are valid for all partition types, and are especially useful for interval partitions. When you insert data into a table, interval partitions are automatically created as needed.

For each `partition_key_value` or `subpartition_key_value` of the preceding elements, a value is specified for each partition key column. If multiple columns are used as partition keys, you must specify a value for each partition key. If a composite partitioning scheme is used, you must specify a value for each subpartition key after you specify a value for a partition key. Separate all partition key values with commas (,). If an interval partitioning scheme is used, you can specify only one `partition_key_value`. The specified value must be a valid value of the `NUMBER` or a date and time data type. Your SQL statements operate on the partitions or subpartitions that contain the specified values.

Limits on extended names

The use of partition-extended and subpartition-extended table names is subject to the following limits:

- No remote tables in names: Partition-extended and subpartition-extended table names cannot contain database links or synonyms that can be translated to tables that contain database links. To use remote partitions and subpartitions, create a view at the remote site by using the extended table name syntax and then reference the remote view.
- No synonyms in names: You must use base tables to specify partition-extended and subpartition-extended names. You cannot use synonyms, views, or other objects.
- In the `PARTITION FOR` and `SUBPARTITION FOR` clauses, you cannot specify the `DEFAULT` or `MAXVALUE` keyword, or a bind variable for `partition_key_value` or `subpartition_key_value`.
- In the `PARTITION` and `SUBPARTITION` clauses, you cannot specify a bind variable for the partition or subpartition name.

Examples

In the following example, **sales** is a partitioned table with the **sales_q1_2000** partition. You can execute the following statement to create a view of the **sales_q1_2000** partition, and then use the view in the same way as you use a table. In this example, some rows are deleted from the partition.

```
/*Create the Q1_2000_sales view for the sales_q1_2000 partition.*/
CREATE VIEW Q1_2000_sales
AS
SELECT *
FROM sales PARTITION (SALES_Q1_2000);
/*Delete the record in which the value in the amount_sold column is less than 0 from the Q1_2000_sale
s view.*/
DELETE FROM Q1_2000_sales
WHERE amount_sold < 0;
```

17.1.5.4.9.5. Reference object type attributes and methods

To reference object type attributes or methods in an SQL statement, qualify the reference by using a table alias. In the following example, the sample schema **ob** contains the **cust_address_typ** type and the **customers** table, and the **customers** table has a **cust_address_typ** column named **cust_address**.

```
CREATE TYPE cust_address_typ
  OID '82A4AF6A4CD1656DE034080020E0EE3D'
AS OBJECT
    (street_address    VARCHAR2(40),
     postal_code       VARCHAR2(10),
     city              VARCHAR2(30),
     state_province    VARCHAR2(10),
     country_id        CHAR(2));
/
CREATE TABLE customers
  (customer_id        NUMBER(6),
   cust_first_name    VARCHAR2(20) CONSTRAINT cust_fname_nn NOT NULL,
   cust_last_name     VARCHAR2(20) CONSTRAINT cust_lname_nn NOT NULL,
   cust_address       cust_address_typ,
. . .
```

In an SQL statement, you must qualify the reference to the **postal_code** attribute by using a table alias, as shown in the following example:

```
SELECT c.cust_address.postal_code
FROM customers c;

UPDATE customers c
SET c.cust_address.postal_code = '610000'
WHERE c.cust_address.city = 'chengdu'
AND c.cust_address.state_province = 'SICHUAN';
```

To reference a member method that does not accept arguments, provide empty parentheses. For example, the **ob** sample schema contains the **category_tab** object table based on **catalog_typ**, and this table contains the `getCatalogName` member function. To call this method in an SQL statement, provide empty parentheses, as shown in the following example:

```
SELECT TREAT(VALUE(c) AS catalog_typ).getCatalogName() "Catalog Type"
FROM categories_tab c
WHERE category_id = 10;
```

The following result is returned:

```
+----------------+
| Catalog Type   |
+----------------+
| online catalog |
+----------------+
```

17.1.5.5. Operators

17.1.5.5.1. Operator overview

Operators generally connect individual data items, such as operands or parameters, and return the results. In terms of syntax, an operator can appear before or after an operand, or between two operands. Special characters or keywords are generally used to represent operators. For example, slashes (/) are used to represent division operators. This topic discusses the operators that excludes logical or BOOLEAN operators. You cannot use these operators as the `WHERE` or `HAVING` conditions for queries or subqueries.

This topic describes the following operators in detail:

- Arithmetic operators
- Concatenation operators
- Hierarchical query operators
- Set operators

Unary and binary operators

Operators can be divided into two categories:

- Unary operators: A unary operator performs an operation on only one operand. The following general format is used for unary operators:

```
operator operand
```

- Binary operators: A binary operator performs an operation on two operands. The following general format is used for binary operators:

```
operand 1 operator operand 2
```

Other operators in the special formats can accept more than two operands. If a NULL operand is provided for an operator, the result is always null. The only operator that does not comply with this rule is the concatenation (||) operator.

Operator precedences

The precedences affect the order in which ApsaraDB for OceanBase calculates different operators in the same expression. When an expression that contains multiple operators is calculated, ApsaraDB for OceanBase first calculates the operator that has a higher precedence and then calculates the operator that has a lower precedence. If the precedences of two operators are equal, the operators that have the same precedence in the expression are calculated from left to right.

The following table lists the precedences of SQL operators from highest to lowest. Operators that are listed on the same line have the same precedence.

Operator	Operation method
+, - (it is used as an unary operator), PRIOR, and CONNECT_BY_ROOT	Plus sign, unary minus, and the position in a hierarchy.
* and /	Multiplication and division.
+, - (it is used as a binary operator), and ||	Addition, subtraction, and concatenation.
After the database evaluates operators, it evaluates case expressions.	For more information, see SQL condition overview.

In the following example, multiplication takes precedence over addition. The database first multiplies 2 by 3 and then adds 1 to the result.

```
1+2*3
```

You can also use parentheses in an expression to limit operator precedences. ApsaraDB for OceanBase evaluates the expressions that are enclosed in parentheses, and then evaluates the expressions that are outside the parentheses.

SQL also supports the set operators `UNION` , `UNION ALL` , `INTERSECT` , and `MINUS` . Set operators merge rowsets that are returned by queries instead of individual data items. All the set operators have equal precedence.

17.1.5.5.2. Arithmetic operators

Arithmetic operators perform operations, such as negation, addition, subtraction, multiplication, and division, on one or two parameters. Some of arithmetic operators are also used to calculate date and time and interval values. The parameters of arithmetic operators must be resolved to the numeric data type or a data type that can be directly converted into the numeric data type by the database.

Unary arithmetic operators return the same data type as the data type of the parameters. For binary arithmetic operators, ApsaraDB for OceanBase determines the parameter that has the highest numeric precedence in the expression, and then converts the remaining parameters to the data type of this parameter.

The following table describes arithmetic operators.

Operator	Description
+, -	If the operators represent positive and negative, they are unary operators.
+, -	If the operators represent addition and subtraction operations, they are binary operators.
*, /	The binary operators. They represent multiplication and division operations.

Two consecutive minus signs (--) cannot be used in arithmetic expressions to indicate double negation or the subtraction of a negative value. This is because the characters -- are used to specify comments in SQL statements. You can separate two consecutive minus signs with spaces or parentheses. For more information about the comments in SQL statements, see Comments.

The following example shows an SQL query that uses the + and - operators to represent positive and negative:

```
SELECT * FROM order_items WHERE quantity = -1 ORDER BY order_id, line_item_id, product_id;
SELECT * FROM employees WHERE -salary < 0 ORDER BY employee_id;
```

The following example shows an SQL query that uses the + and - operators to represent addition and subtraction operations:

```
SELECT hire_date FROM employees WHERE SYSDATE - hire_date > 365 ORDER BY hire_date;
```

The following example shows an SQL query that uses the * and / operators to represent multiplication and division operations:

```
UPDATE employees SET salary = salary * 1.1;
UPDATE employees SET salary = salary / 2;
```

17.1.5.5.3. Concatenation operators

The concatenation operator || is used to concatenate strings and data of the `CLOB` type.

After two strings are concatenated, another string is generated. If the data types of the two strings are `CHAR`, the returned result is of the `CHAR` data type and can be at most 2,000 characters in length. If the data type of one of the concatenated strings is `VARCHAR2`, the returned result is of the `VARCHAR2` data type and can be at most 4,000 characters in length. If one of the parameters on the two sides of the concatenation operator is a value of the `CLOB` data type, the returned result is of a temporary `CLOB` data type. The trailing spaces in the strings are preserved by concatenation, regardless of the data types of the parameters on both sides of the concatenation operator.

Although ApsaraDB for OceanBase regards a zero-length string as NULL, an operand that contains values is generated after an operand that contains values is concatenated with another zero-length string. Therefore, NULL can be generated only if two NULL strings are concatenated. However, this may not continue to be true in future ApsaraDB for OceanBase database versions. To concatenate expressions that may be NULL, use the `NVL` function to directly convert the expressions to zero-length strings.

In the following example, a table that contains columns of the `CHAR` and `VARCHAR2` types is created. Values that have and do not have trailing spaces are inserted and concatenated.

```
CREATE TABLE tab1 (col1 VARCHAR2(6), col2 CHAR(10), col3 VARCHAR2(10), col4 CHAR(6));
INSERT INTO tab1 (col1, col2, col3, col4) VALUES ('abc', 'def    ', 'ghi    ', 'jkl');
SELECT col1 || col2 || col3 || col4 "Concatenation" FROM tab1;
```

The following result is returned:

```
+----------------------------+
| Concatenation              |
+----------------------------+
| abcdef       ghi    jkl    |
+----------------------------+
```

17.1.5.5.4. Hierarchical query operators

The `PRIOR` and `CONNECT_BY_ROOT` operators are valid in only hierarchical queries.

PRIOR operator

In a hierarchical query, the `CONNECT BY` condition must contain at least one expression that is qualified by the `PRIOR` operator. The result of the `PRIOR` operator is obtained by using the parent row of the current row to compute the expression that immediately follows the operator.

`PRIOR` is most commonly used in an equality comparison of columns. `PRIOR` can be placed on either side of the comparison operator. Theoretically, operators other than the equal sign (=) can be used in `CONNECT BY` clauses. However, the conditions that are created by these other operators may cause loops. In this case, ApsaraDB for OceanBase detects the loops at runtime and returns an error.

CONNECT_BY_ROOT operator

`CONNECT_BY_ROOT` is a unary operator and is valid in only hierarchical queries. The parameters of `CONNECT_BY_ROOT` can be columns and most expressions. The result of the `CONNECT_BY_ROOT` operator equals to the value that is obtained by using the data of the root row to calculate the parameter.

You cannot specify this operator in a `START WITH` or `CONNECT BY` condition.

17.1.5.5.5. Set operators

A set operator merges two query results into one result. The queries that contain set operators are compound queries.

Operator	Return value
UNION	Returns all the distinct rows that are selected by a query.
UNION ALL	Returns all the rows that are selected by a query. The return result includes all the duplicates.
INTERSECT	Returns all the distinct rows that are selected by both queries.
MINUS	Returns all the distinct rows that are selected by the first query. The return result excludes the rows that are included in the result of the second query.

For more information and examples about the set operators, see the descriptions of the UNION ALL, INTERSECT, and MINUS operators in the Queries and subqueries topic.

17.1.5.5.6. Collations

Collations (COLLATE operators) allow you to specify a collation and reset the collation of the character type or specify the collation of a single expression.

COLLATE is a postfix unary operator. It has the same precedence as other unary operators, but it is executed after all the prefix unary operators are executed. The COLLATE operator can be applied to the expressions of the VARCHAR2, CHAR, LONG, NVARCHAR, or NCHAR type.

The collation_name parameter that follows the COLLATE operator is the collation that is to be specified or used. When the collation name includes spaces, you must enclose the name in double quotation marks ('').

Example:

```
obclient> create table t1(c1 varchar(20));
obclient> insert into t1 values('a');
obclient> insert into t1 values('b');
obclient> insert into t1 values(1);
obclient> select * from t1 order by c1 collate utf8mb4_general_ci;
+------+
| C1   |
+------+
| 1    |
| a    |
| b    |
+------+
3 rows in set (0.00 sec)
```

The following collations are supported:

```
obclient> show collation;
+--------------------+---------+-----+---------+----------+---------+
| COLLATION          | CHARSET | ID  | DEFAULT | COMPILED | SORTLEN |
+--------------------+---------+-----+---------+----------+---------+
| utf8mb4_general_ci | utf8mb4 |  45 | Yes     | Yes      |       1 |
| utf8mb4_bin        | utf8mb4 |  46 | NULL    | Yes      |       1 |
| binary             | binary  |  63 | Yes     | Yes      |       1 |
| gbk_chinese_ci     | gbk     |  28 | Yes     | Yes      |       1 |
| gbk_bin            | gbk     |  87 | NULL    | Yes      |       1 |
| utf16_general_ci   | utf16   |  54 | Yes     | Yes      |       1 |
| utf16_bin          | utf16   |  55 | NULL    | Yes      |       1 |
| utf8mb4_unicode_ci | utf8mb4 | 224 | NULL    | Yes      |       1 |
| utf16_unicode_ci   | utf16   | 101 | NULL    | Yes      |       1 |
| gb18030_chinese_ci | gb18030 | 248 | Yes     | Yes      |       1 |
| gb18030_bin        | gb18030 | 249 | NULL    | Yes      |       1 |
+--------------------+---------+-----+---------+----------+---------+
11 rows in set (0.01 sec)
```

17.1.5.6. Functions

17.1.5.6.1. Function overview

Functions work in a similar way as operators. After data elements are passed to a function as the input of parameters, the function returns a result. However, functions differ from operators in the format of input parameters. Functions can contain a variable number of parameters. A function can contain one, two, or more parameters.

```
Function name (parameter, parameter, ...)
```

A function without parameters is similar to a pseudocolumn. For more information, see Pseudo columns. A pseudocolumn returns a different value for each row in the result set. However, a function without variables returns the same value for all rows.

About functions

Built-in functions of ApsaraDB for OceanBase can be used in SQL statements. An expected data type is specified for each parameter of a function. If the data type of an input argument is expected, ApsaraDB for OceanBase attempts to convert the data type to the expected one and then executes the SQL statement.

Null values in functions

If the input argument is NULL, most functions return NULL as the result. In this case, you can use the `NVL` function to return a non-null value. For example, assume that a table is created to store commission data and the **commission_pct** column stores the commission values. If the value in the commission_pct column is NULL, the `NVL(commission_pct,0)` expression returns 0. If the value in the **commission_pct** column is not NULL, the expression returns the actual commission value.

Function types and lists

In the following function lists, each parameter or return value of each function has a specific data type.

> **Notice**
>
> When you apply a function to a `LOB` column in an SQL statement, ApsaraDB for OceanBase creates a temporary `LOB` column during SQL processing or PL/SQL processing. Some limits are set for using the column. For more information, see Compatibility with Oracle databases.

This topic describes two categories of functions:

- **Single-row functions**: include numeric functions, string functions that return strings, string functions that return numbers, date and time functions, general comparison functions, conversion functions, encoding and decoding functions, and null-related functions.
- **Statistical functions**: include aggregate functions and analytic functions.

A single-row function returns a result value for each row of a queried table or view. These functions can be used in SQL clauses such as `SELECT`, `WHERE`, `START WITH`, `CONNECT BY`, and `HAVING`.

Aggregate functions and analytic functions perform aggregate calculations on a group of rows. An aggregate function returns only a single row for each group of rows. However, an analytic function returns multiple rows for each group of rows. A group of rows is also called a window. In most cases, aggregate functions are used in combination with the `GROUP BY` clause in the `SELECT` statement. When you use an aggregate function, the database divides the rows of a queried table or view into groups and applies the aggregate function to each group of rows. Then, one result row is returned for each group of rows.

When you use analytic functions, you must use the special keyword `OVER` to specify a window. For more information about window functions, see Window functions.

Numeric functions

Numeric functions accept numeric values as the input and return numeric values. Most numeric functions return values of

the `NUMBER` data type. The return values can be accurate to 38 decimal places. Some functions are related to advanced algebra. Some of these functions, such as `COS`, `COSH`, `EXP`, `LN`, `LOG`, `SIN`, `SINH`, `SQRT`, `TAN`, and `TANH`, return values that are accurate to 36 decimal points. Other functions that are related to advanced algebra, such as `ACOS`, `ASIN`, `ATAN`, and `ATAN2`, return values that are accurate to 30 decimal places.

Function category	Function subcategory	Function name	Description
Single-row function	Numeric function	ABS	A mathematical function that returns the absolute (positive) value of the specified numeric expression.

Function category	Function subcategory	Function name	Description
Single-row function	Numeric function	ACOS	Returns the angle that is represented in radians, whose cosine is specified by using the `NUMBER` expression. This angle is also called arccosine.
Single-row function	Numeric function	ASIN	ApsaraDB for OceanBase does not support this function.
Single-row function	Numeric function	ATAN	ApsaraDB for OceanBase does not support this function.
Single-row function	Numeric function	ATAN2	ApsaraDB for OceanBase does not support this function.
Single-row function	Numeric function	BITAND	Performs a bitwise operation by using the bitwise AND operator. Both the input and output values are of the same `INT` type.
Single-row function	Numeric function	CEIL	Returns the smallest integer that is greater than or equal to the value of `numeric_expression` .
Single-row function	Numeric function	COS	ApsaraDB for OceanBase does not support this function.
Single-row function	Numeric function	COSH	ApsaraDB for OceanBase does not support this function.

Function category	Function subcategory	Function name	Description
Single-row function	Numeric function	EXP	Returns e raised to the power of numeric_expression.
Single-row function	Numeric function	FLOOR	Returns the largest integer that is equal to or less than the value of `numeric_expression`.
Single-row function	Numeric function	LN	Returns the e-based logarithm of `numeric_expression`.
Single-row function	Numeric function	LOG	Returns the `x`-based logarithm of `y`.
Single-row function	Numeric function	MOD	Returns the remainder that is produced when `x` is divided by `y`.
Single-row function	Numeric function	POWER	Returns `x` raised to the power of `y`.
Single-row function	Numeric function	REMAINDER	Returns the remainder that is produced when `x` is divided by `y`.
Single-row function	Numeric function	ROUND	Returns the rounded value of `numeric`.

Function category	Function subcategory	Function name	Description
Single-row function	Numeric function	SIGN	Returns the sign of the number `n`. This function returns 1 if n is greater than 0, returns -1 if n is less than 0, and returns 0 if n is equal to 0.
Single-row function	Numeric function	SIN	ApsaraDB for OceanBase does not support this function.
Single-row function	Numeric function	SINH	ApsaraDB for OceanBase does not support this function.
Single-row function	Numeric function	SQRT	Returns the square root of `n`.
Single-row function	Numeric function	TAN	ApsaraDB for OceanBase does not support this function.
Single-row function	Numeric function	TANH	ApsaraDB for OceanBase does not support this function.
Single-row function	Numeric function	TRUNC	Truncates `numeric` to the specified `precision` and returns the result value.
Single-row function	Numeric function	WIDTH_BUCKET	ApsaraDB for OceanBase does not support this function.

String functions that return strings

The maximum length of the value returned by a function varies based on the data type of the value. For example, assume that a function returns a `VARCHAR2` value whose length exceeds the maximum length of a `VARCHAR2` value. In this case, ApsaraDB for OceanBase truncates the value and returns the truncated value without displaying a prompt on the client.

Notice

If a function returns a `CLOB` value whose length exceeds the maximum length of a CLOB value, ApsaraDB for OceanBase returns no data and displays an error message.

Function category	Function subcategory	Function name	Description
Single-row function	String functions that return strings	CHR	Converts `n` to one or more equivalent characters. The return value is related to the current system character set.
Single-row function	String functions that return strings	CONCAT	Concatenates two strings.
Single-row function	String functions that return strings	INITCAP	Returns a string in which the first letter of each word is uppercase and other letters are lowercase.
Single-row function	String functions that return strings	LOWER	Converts all letters in a string to lowercase letters.
Single-row function	String functions that return strings	LPAD	Left-pads the `c1` string to the given length `n` with the `c2` string.
Single-row function	String functions that return strings	LTRIM	Trims the string that appears on the left.
Single-row function	String functions that return strings	REGEXP_REPLACE	Replaces an object with a regular expression.
Single-row function	String functions that return strings	REGEXP_SUBSTR	ApsaraDB for OceanBase does not support this function.
Single-row function	String functions that return strings	REPLACE	Replaces a specified string with a new string in a character expression.

Function category	Function subcategory	Function name	Description
Single-row function	String functions that return strings	RPAD	Right-pads the `c1` string to the given length `n` with the `c2` string.
Single-row function	String functions that return strings	RTRIM	Trims a string that appears on the right. This function is useful for formatting the output of a query.
Single-row function	String functions that return strings	SUBSTR	Extracts a substring. A multibyte character, such as a full-width character, is counted as one character.
Single-row function	String functions that return strings	TRANSLATE	Replaces a specified character with a new character in a character expression. A multibyte character, such as a full-width character, is counted as one character.
Single-row function	String functions that return strings	TRIM	Trims leading characters, trailing characters, or both of them from a string.
Single-row function	String functions that return strings	UPPER	Converts all letters in a string to uppercase letters.

String functions that return numbers

Function category	Function subcategory	Function name	Description
Single-row function	String functions that return numbers	ASCII	Returns the ASCII value of the leftmost character in a character expression.

Single-row function	String functions that return numbers	INSTR	Searches for a specified character in a string, and returns the position where the specified character is found.
Single-row function	String functions that return numbers	LENGTH	Returns the length of a string.
Single-row function	String functions that return numbers	REGEXP_COUNT	ApsaraDB for OceanBase does not support this function.
Single-row function	String functions that return numbers	REGEXP_INSTR	ApsaraDB for OceanBase does not support this function.

Date and time functions

Date and time functions support input parameters of the following data types: date-related data types (`DATE`), timestamp-related data types (`TIMESTAMP` , `TIMESTAMP WITH TIME ZONE` , and `TIMESTAMP WITH LOCAL TIME ZONE`), and interval-related data types (`INTERVAL DAY TO SECOND` and `INTERVAL YEAR TO MONTH`).

The following functions support input parameters of only the `DATE` data type: `ADD_MONTHS` , `CURRENT_DATE, LAST_DAY` , `NEW_TIME` , and `NEXT_DAY` .

Assume that you attempt to insert a `TIMESTAMP` value as an argument into the preceding functions. ApsaraDB for OceanBase implicitly converts the data type, passes the converted data to the function for calculations, and then returns a `DATE` value.

> **Notice**
>
> `MONTHS_BETWEEN` returns a number.
>
> `ROUND` and `TRUNC` cannot implicitly convert data types and support input parameter of only the `DATE` type. If you pass values of other data types to these functions, errors are reported.

The other date and time functions support input parameters of the three data types and return values of the same data types as the input parameters.

Function category	Function subcategory	Function name	Description
Single-row function	Date and time function	ADD_MONTHS	Returns the date value that is `n` months after `date`. If `n` is a negative number, this function returns the date value that is `n` months before `date`.
Single-row function	Date and time function	CURRENT_DATE	Returns the current date in the session time zone.
Single-row function	Date and time function	CURRENT_TIMESTAMP	Returns a value of the `TIMESTAMP WITH TIME ZONE` data type. The return value indicates the current date in the session time zone and contains the information of the current time zone.
Single-row function	Date and time function	DBTIMEZONE	Returns the time zone of the current database instance. In ApsaraDB for OceanBase, the database time zone is UTC+0 and cannot be changed.
Single-row function	Date and time function	EXTRACT (datetime)	Extracts elements such as the year, month, day, hour, minute, and second values from a specified time field or expression.
Single-row function	Date and time function	FROM_TZ	Combines a value of the `TIMESTAMP` data type with the time zone information into a time value of the `TIMESTAMP WITH TIME ZONE` data type.

Function category	Function subcategory	Function name	Description
Single-row function	Date and time function	LAST_DAY	Returns the date of the last day of the month in which the specified `date` falls.
Single-row function	Date and time function	LOCALTIMESTAMP	Returns a value of the `TIMESTAMP` data type. The return value indicates the current date in the session time zone.
Single-row function	Date and time function	MONTHS_BETWEEN	Returns the number of months between `date1` and `date2`.
Single-row function	Date and time function	NEW_TIME	ApsaraDB for OceanBase does not support this function.
Single-row function	Date and time function	NEXT_DAY	Returns the date value of the weekday `c1` in the week following `d1`.
Single-row function	Date and time function	NUMTODSINTERVAL	Converts the argument `n` to a value of the `INTERVAL DAY TO SECOND` data type. The `interval_unit` parameter specifies the measurement unit.
Single-row function	Date and time function	NUMTOYMINTERVAL	Converts the argument `n` to a value of the `INTERVAL YEAR TO MONTH` data type. The `interval_unit` parameter specifies the measurement unit.

Function category	Function subcategory	Function name	Description
Single-row function	Date and time function	ROUND (date)	Returns a date and time value that is nearest to the specified `date`. The `fmt` parameter specifies the unit that is used to measure the interval between the returned date and the specified date.
Single-row function	Date and time function	SESSIONTIMEZONE	Returns the time zone of the current session.
Single-row function	Date and time function	SYS_EXTRACT_UTC	Returns the UTC time that corresponds to the specified time.
Single-row function	Date and time function	SYSDATE	Returns the current date.
Single-row function	Date and time function	SYSTIMESTAMP	Returns the current system date which contains the current time zone information. Six digits appear after the decimal point of the second value.
Single-row function	Date and time function	TO_CHAR (datetime)	Converts a value of the `DATE`, `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE`, `TIMESTAMP WITH LOCAL TIME ZONE`, `INTERVAL DAY TO SECOND`, or `INTERVAL YEAR TO MONTH` data type to a value of the `VARCHAR2` data type. The `fmt` parameter specifies the format of the return value.

Function category	Function subcategory	Function name	Description
Single-row function	Date and time function	TO_DSINTERVAL	Converts a string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type to a value of the `INTERVAL DAY TO SECOND` data type. You can perform the addition and subtraction arithmetic operations on date and time values by using this function.
Single-row function	Date and time function	TO_TIMESTAMP	Converts a string to a value of the `TIMESTAMP` data type.
Single-row function	Date and time function	TO_TIMESTAMP_TZ	Converts a string to a value of the `TIMESTAMP WITH TIME ZONE` data type, which contains the time zone information.
Single-row function	Date and time function	TO_YMINTERVAL	Converts a string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type to a value of the `INTERVAL YEAR TO MONTH` data type. You can perform the addition and subtraction arithmetic operations on date and time values by using this function.

Function category	Function subcategory	Function name	Description
Single-row function	Date and time function	TRUNC (date)	Returns a date and time value that is nearest to the specified `date`. The `fmt` parameter specifies the unit in which the interval between the return value and the specified date is measured. The returned date value precedes `date`.
Single-row function	Date and time function	TZ_OFFSET	Returns the offset of the `n` time zone. The time zone offset is the difference between a time zone and the UTC+0 time zone in hours and minutes.

General comparison functions

You can use this type of functions to locate the maximum and minimum values in a set of values in a short time.

Function category	Function subcategory	Function name	Description
Single-row function	General comparison function	GREATEST	Returns the maximum value in a list of one or more expressions.
Single-row function	General comparison function	LEAST	Returns the minimum value in a list of one or more expressions.

Conversion functions

You can convert a value from one data type to another data type by using this type of functions.

Function category	Function subcategory	Function name	Description
Function category	**Function subcategory**	**Function name**	**Description**
Single-row function	Conversion function	ASCIISTR	ApsaraDB for OceanBase does not support this function.

Function category	Function subcategory	Function name	Description
Single-row function	Conversion function	BIN_TO_NUM	ApsaraDB for OceanBase does not support this function.
Single-row function	Conversion function	CHARTOROWID	ApsaraDB for OceanBase does not support this function.
Single-row function	Conversion function	HEXTORAW	Converts a character that contains hexadecimal numbers and is of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type to a value of the RAW data type.
Single-row function	Conversion function	RAWTOHEX	Converts a binary number to a hexadecimal format string.
Single-row function	Conversion function	TO_BINARY_DOUBLE	Returns a 64-bit double-precision floating-point number.
Single-row function	Conversion function	TO_BINARY_FLOAT	Returns a 32-bit single-precision floating-point number.
Single-row function	Conversion function	TO_CHAR (character)	Converts an `NCHAR`, `NVARCHAR2`, or `CLOB` character to a database character set.

Function category	Function subcategory	Function name	Description
Single-row function	Conversion function	TO_CHAR (datetime)	Converts a value of the `DATE`, `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE`, `TIMESTAMP WITH LOCAL TIME ZONE`, `INTERVAL DAY TO SECOND`, or `INTERVAL YEAR TO MONTH` data type to a value of the `VARCHAR2` data type. The `fmt` parameter specifies the format of the return value.
Single-row function	Conversion function	TO_CHAR (number)	Converts the value `n` of the `NUMBER`, `BINARY_FLOAT`, or `BINARY_DOUBLE` data type to a value of the `VARCHAR2` data type in the format specified by `fmt`.
Single-row function	Conversion function	TO_DATE	Converts a character of the `CHAR`, `VARCHAR`, `NCHAR`, or `NVARCHAR2` data type to a value of the DATE data type.

Function category	Function subcategory	Function name	Description
Single-row function	Conversion function	TO_DSINTERVAL	Converts a string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type to a value of the `INTERVAL DAY TO SECOND` data type. You can perform the addition and subtraction arithmetic operations on a date and time value by using this function.
Single-row function	Conversion function	TO_NUMBER	Converts `expr` to a numeric value.
Single-row function	Conversion function	TO_TIMESTAMP	Converts a string to a value of the `TIMESTAMP` data type.
Single-row function	Conversion function	TO_TIMESTAMP_TZ	Converts a string to a value of the `TIMESTAMP WITH TIME ZONE` data type, which contains the time zone information.

Function category	Function subcategory	Function name	Description
Single-row function	Conversion function	TO_YMINTERVAL	Converts a string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type to a value of the `INTERVAL YEAR TO MONTH` data type. You can perform the addition and subtraction arithmetic operations on a date and time value by using this function.

Encoding and decoding functions

You can encode and decode data in ApsaraDB for OceanBase by using this type of functions.

Function category	Function subcategory	Function name	Description
Single-row function	Encoding and decoding function	DECODE	Returns a value that matches the specified conditions.
Single-row function	Encoding and decoding function	ORA_HASH	Retrieves the hash value of a corresponding expression.
Single-row function	Encoding and decoding function	VSIZE	Returns the size of `X` in bytes.

Null value-related functions

Function category	Function subcategory	Function name	Description
Single-row function	Null value-related function	COALESCE	Returns the first non-null expression in a parameter list. You must specify at least two parameters.

Function category	Function subcategory	Function name	Description
Single-row function	Null value-related function	LNNVL	Determines whether one or two operands in a condition are `NULL`.
Single-row function	Null value-related function	NULLIF	ApsaraDB for OceanBase does not support this function.
Single-row function	Null value-related function	NVL	Returns a non-NULL value from two expressions. If `expr1` and `expr2` return NULL, the NVL function returns NULL.
Single-row function	Null value-related function	NVL2	Returns different values based on whether an expression is null. If `expr1` is not null, this function returns the value of `expr2`. If `expr1` is null, this function returns the value of `expr3`. If `expr2` and `expr3` are of different data types, this function converts the data type of `expr3` to that of `expr1`.

Environment-related functions

Functions of this type provide the environment information about sessions or tenants.

Function category	Function subcategory	Function name	Description
Single-row function	Environment-related function	SYS_CONTEXT	ApsaraDB for OceanBase does not support this function.
Single-row function	Environment-related function	UID	ApsaraDB for OceanBase does not support this function.

Function category	Function subcategory	Function name	Description
Single-row function	Environment-related function	USER	ApsaraDB for OceanBase does not support this function.

Aggregate functions

Function category	Function subcategory	Function name	Description
Statistical function	Aggregate function	AVG	Returns the average value of all values in a numeric column.
Statistical function	Aggregate function	COUNT	Queries the number of rows for the `expr` parameter.
Statistical function	Aggregate function	SUM	Returns the sum of all values in a specified column.
Statistical function	Aggregate function	GROUPING	ApsaraDB for OceanBase does not support this function.
Statistical function	Aggregate function	MAX	Returns the maximum value of a specified column.
Statistical function	Aggregate function	MIN	Returns the minimum value of a specified column.
Statistical function	Aggregate function	LISTAGG	Converts a column to a row. The `LISTAGG` function sorts the data within each group that is specified in the `ORDER BY` clause and merges the values in the measure column.

Function category	Function subcategory	Function name	Description
Statistical function	Aggregate function	ROLLUP	Returns a subtotal for each group and a grand total for all groups during data statistical analysis and report generation.
Statistical function	Aggregate function	STDDEV	Calculates the population standard deviation.
Statistical function	Aggregate function	STDDEV_POP	Calculates the population standard deviation.
Statistical function	Aggregate function	STDDEV_SAMP	Calculates the sample standard deviation.
Statistical function	Aggregate function	VARIANCE	Returns the variance of a specified column.
Statistical function	Aggregate function	APPROX_COUNT_DISTINCT	Calculates the number of rows in a column where duplicates are removed, and can return only an approximate value. You can use this function to further calculate the selectivity of the referenced column.

Analytic functions

Function category	Function subcategory	Function name	Description
Statistical function	Analytic function	AVG	Returns the average value of all values in a numeric column.
Statistical function	Analytic function	COUNT	Queries the number of rows for the `expr` parameter.

Function category	Function subcategory	Function name	Description
Statistical function	Analytic function	CUME_DIST	Calculates the cumulative distribution of a value in a group of values.
Statistical function	Analytic function	DENSE_RANK	Calculates the rank of a row in an ordered group of rows and returns the rank as `NUMBER`.
Statistical function	Analytic function	MAX	Returns the maximum value of a specified column.
Statistical function	Analytic function	MIN	Returns the minimum value of a specified column.
Statistical function	Analytic function	SUM	Returns the sum of all values in a specified column.
Statistical function	Analytic function	FIRST_VALUE	Returns the first value in a set of ordered values.
Statistical function	Analytic function	LAG	Provides access to a multi-row table without a self join.
Statistical function	Analytic function	LAST_VALUE	Returns the last value in a set of ordered values.
Statistical function	Analytic function	LEAD	Provides access to multiple rows of a table without a self join. Given a set of rows returned from a query and the position of the cursor, `LEAD` provides access to a row at a given physical offset beyond that position.
Statistical function	Analytic function	LISTAGG	Converts a column to a row.

Function category	Function subcategory	Function name	Description
Statistical function	Analytic function	NTH_VALUE	Returns the value of `measure_expr` in the `n` th row of the window defined by `analytic_clause`.
Statistical function	Analytic function	NTILE	Divides an ordered dataset into several buckets and assigns an appropriate bucket number to each row. `expr` specifies the number of buckets.
Statistical function	Analytic function	PERCENT_RANK	This function is similar to the `CUME_DIST` function that calculates the cumulative distribution. The return value ranges from 0 to 1. The `PERCENT_RANK` function of the first row in a set is 0. The return value is NUMBER.
Statistical function	Analytic function	SUM	Returns the sum of all values in a specified column.
Statistical function	Analytic function	RANK	Determines the rank of a group of values based on the `ORDER BY` expression in the `OVER` clause.
Statistical function	Analytic function	RATIO_TO_REPORT	Calculates the ratio of a value to the sum of a group of values.
Statistical function	Analytic function	ROW_NUMBER	Assigns a unique number to each row to which the function is applied.

Function category	Function subcategory	Function name	Description
Statistical function	Analytic function	STDDEV	Calculates the population standard deviation.
Statistical function	Analytic function	STDDEV_POP	Calculates the population standard deviation.
Statistical function	Analytic function	STDDEV_SAMP	Calculates the sample standard deviation.
Statistical function	Analytic function	VARIANCE	Returns the variance of a specified column.

More information

For more information about the `OVER` keyword of analytic functions, see Window function description.

17.1.5.6.2. Single-row functions

17.1.5.6.2.1. Numeric functions

ABS

The `ABS` function is a mathematical function that returns the absolute value of the specified numeric expression. The absolute value is a positive value. `ABS` changes negative values to positive values and does not affect zero or positive values.

Syntax

```
ABS (numeric_expression)
```

Parameters

Parameter	Description
numeric_expression	The expression of the exact numeric data types or the approximate numeric data types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`.

Return type

The return type is the same as the data type of `numeric_expression`.

Examples

This example shows the results of using the `ABS` function for three different numbers.

Execute the following statement:

```
SELECT ABS(-1.0), ABS(0.0), ABS(1.0), ABS(1.666) FROM DUAL;
```

The following query result is returned:

```
+-----------+----------+----------+------------+
| ABS(-1.0) | ABS(0.0) | ABS(1.0) | ABS(1.666) |
+-----------+----------+----------+------------+
|         1 |        0 |        1 |      1.666 |
+-----------+----------+----------+------------+
```

ACOS

The `ACOS` function returns an angle that is expressed in radians. The cosine of the angle is a specified `NUMBER` expression. It is also called the arc cosine.

Syntax

```
ACOS (num_expression)
```

Parameters

Parameter	Value	Description
num_expression	-1.00~1.00	The expression of the `NUMBER` type or the types that can be implicitly converted to the `NUMBER` type. Only the values from -1.00 to 1.00 are valid. For values beyond this range, the function returns `NULL` and reports an error.

Return type

`NUMBER` data type

Examples

This example returns the value of `ACOS` for the specified number.

Execute the following statement:

```
SELECT ACOS(0.3)"acos_test" FROM DUAL;
```

The following query result is returned:

```
+-----------+
| acos_test |
+-----------+
|1.26610367 |
+-----------+
```

BITAND

The operator performs a bitwise AND operation. The input and output types are the `NUMBER` data type.

Syntax

```
BITAND (nExpression1, nExpression2)
```

Parameters

Parameter	Description
nExpression1, nExpression2	Specifies two numeric values for the bitwise `AND` operation. If `nExpression1` and `nExpression2` are not the integer type, they are converted to integers before the bitwise `AND` operation is performed.

Return type

`NUMBER` data type

Examples

a is 0100 in binary.

b is 0110 in binary.

Compare the binary numbers. If the digits in corresponding positions are all 1, the value of this bit is 1. Otherwise, the value is 0. Therefore, the result is BITAND(2 , 3) = 0100 = 2.

Execute the following statement:

```
SELECT BITAND(2,3) FROM DUAL;
```

The following query result is returned:

```
+-------------+
| BITAND(2,3) |
+-------------+
|           2 |
+-------------+
```

CEIL

The `CEIL` function returns the smallest integer that is greater than or equal to the numeric value of `numeric_expression` .

Syntax

```
CEIL (numeric_expression)
```

Parameters

Parameter	Description
numeric_expression	The expression of the exact numeric data types or the approximate numeric data types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`.

Return type

The return type is the same as the data type of the `numeric_expression` parameter.

Examples

This example shows the result of using the `CEIL` function for three different numbers.

Execute the following statement:

```
SELECT CEIL(1.2), CEIL(2), CEIL(-12.1) FROM DUAL;
```

The following query result is returned:

```
+-----------+---------+-------------+
| CEIL(1.2) | CEIL(2) | CEIL(-12.1) |
+-----------+---------+-------------+
|         2 |       2 |         -12 |
+-----------+---------+-------------+
```

EXP

The `EXP` function returns e that is raised to the power of `numeric_expression`. e is a mathematical constant and is equal to 2.71828183....

Syntax

```
EXP (numeric_expression)
```

Parameters

Parameter	Description
numeric_expression	The expression of the exact numeric data types or the approximate numeric data types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`.

Return type

The return type is the same as the data type of the `numeric_expression` parameter.

Examples

This example shows the result of querying e that is raised to the power of 4.

Execute the following statement:

```
SELECT EXP(4) "e to the 4th power" FROM DUAL;
```

The following query result is returned:

```
+-------------------------------------------+
| e to the 4th power                        |
+-------------------------------------------+
| 54.59815003314423907811026120286087840279 |
+-------------------------------------------+
```

FLOOR

The `FLOOR` function returns the largest integer that is less than or equal to the numeric value of `numeric_expression` .

Syntax

```
FLOOR (numeric_expression)
```

Parameters

Parameter	Description
numeric_expression	The expression of the exact numeric data types or the approximate numeric data types: `NUMBER` , `FLOAT` , `BINARY_FLOAT` , and `BINARY_DOUBLE` .

Return type

The return type is the same as the data type of the `numeric_expression` parameter.

Examples

This example shows the results of using the `FLOOR` function for three different numbers.

Execute the following statement:

```
SELECT FLOOR(1.2), FLOOR(2), FLOOR(-12.1) FROM DUAL;
```

The following query result is returned:

```
+------------+----------+--------------+
| FLOOR(1.2) | FLOOR(2) | FLOOR(-12.1) |
+------------+----------+--------------+
|          1 |        2 |          -13 |
+------------+----------+--------------+
```

LN

The `LN` function returns the logarithm of `numeric_expression` to the base of e. e is a mathematical constant and is equal to 2.71828183....

Syntax

```
LN (numeric_expression)
```

Parameters

Parameter	Description
numeric_expression	The expression of the exact numeric data types or the approximate numeric data types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`.

Return type

When the data type of the parameter is `BINARY_FLOAT`, the return type is `BINARY_DOUBLE`. In other cases, the return type is the same as the data type of the `numeric_expression` parameter.

Examples

This example shows the result of calculating the logarithm of 4 to the base of e.

Execute the following statement:

```
SELECT LN(4) "Natural log of 4" FROM DUAL;
```

The following query result is returned:

```
+-----------------------------------------+
| Natural log of 4                        |
+-----------------------------------------+
| 1.3862943611198906188344642429163531361 5 |
+-----------------------------------------+
```

LOG

The `LOG` function returns the logarithm of `y` to the base of `x`.

Syntax

```
LOG (x,y)
```

Parameters

Parameter	Description
x,y	The expression of the numeric types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`. `x` and `y` must be greater than 0.

Return type

If the data types of the parameters are `BINARY_FLOAT` and `BINARY_DOUBLE`, the return type is `BINARY_DOUBLE`. In other cases, the return type is `NUMBER`.

Examples

This example shows the logarithm of 8 to the base of 2.

```
SELECT LOG(2,8) FROM DUAL;
```

The following query result is returned:

```
+-----------+
| LOG(2,8) |
+-----------+
|         3 |
+-----------+
```

MOD

The `MOD` function returns the remainder of `x` that is divided by `y`.

> **Notice**
>
> When the `REMAINDER (x,y)` and `MOD (x,y)` functions perform operations, the same formula result=x-y*(x/y) is used. The difference between MOD (x,y) and REMAINDER (x,y) is that the processing methods are different when x/y is calculated. `ROUND(x/y)` is used in the `REMAINDER (x,y)` function, whereas `FLOOR(x/y)` is used in the `MOD (x,y)` function.

Syntax

```
MOD (x,y)
```

Parameters

Parameter	Description
x,y	`x` and `y` are the expressions of the numeric types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`.

Return type

The return type is the same as the data type of the parameter that has higher numeric precedence.

Examples

This example shows the result of calculating the remainder of 23/8 and 24/8.

Execute the following statement:

```
SELECT MOD(23,8), MOD(24,8) FROM DUAL;
```

The following query result is returned:

```
+-----------+-----------+
| MOD(23,8) | MOD(24,8) |
+-----------+-----------+
|         7 |         0 |
+-----------+-----------+
```

POWER

The `POWER` function returns `x` that is raised to the `y` power.

Syntax

```
POWER (x, y)
```

Parameters

Parameter	Description
x,y	`x` and `y` are the expressions of the numeric types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`.

Return type

If the data types of the parameters are `BINARY_FLOAT` and `BINARY_DOUBLE`, the return type is `BINARY_DOUBLE`. In other cases, the return type is `NUMBER`.

Examples

This example shows the results of using the `POWER` function for three different groups of numbers.

Execute the following statement:

```
SELECT POWER(2,2), POWER(1.5,0), POWER(20, -1) FROM DUAL;
```

The following query result is returned:

```
+------------+--------------+--------------+
| POWER(2,2) | POWER(1.5,0) | POWER(20,-1) |
+------------+--------------+--------------+
|          4 |            1 |          .05 |
+------------+--------------+--------------+
```

REMAINDER

The `REMAINDER` function returns the remainder of `x` that is divided by `y`.

> **Notice**
>
> Difference between the REMAINDER function and the MOD function:
>
> When the `REMAINDER (x,y)` and `MOD (x,y)` functions perform operations, the same formula result=x-y*(x/y) is used. The difference is that the processing methods are different when x/y is calculated. `ROUND(x/y)` is used in the `REMAINDER (x,y)` function, whereas `FLOOR(x/y)` is used in the `MOD (x,y)` function. Assume that the decimal part of the value of the x/y parameter in `ROUND(x/y)` is 0.5 in the `REMAINDER` function. If the integer part of the value of x/y is an even number, the value is rounded down to the nearest integer. If the integer part of the value of x/y is an odd number, the value is rounded up the nearest integer. For example, `ROUND(1.5)` is equal to 2, `ROUND(2.5)` is equal to 2, `ROUND(3.5)` is equal to 4, and `ROUND(4.5)` is equal to 4.

Syntax

```
REMAINDER (x, y)
```

Parameter

Parameter	Description
x,y	`x` and `y` are the expressions of the numeric types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`.

Return type

The return type is the same as the data type of the parameter that has higher numeric precedence.

Examples

This example shows the results of using the `MOD` and `REMAINDER` functions to calculate the remainder of 1.5/1. Take note of the difference between the two functions.

Execute the following statement:

```
SELECT MOD(1.5,1), REMAINDER(1.5,1) FROM DUAL;
```

The following query result is returned:

```
+------------+------------------+
| MOD(1.5,1) | REMAINDER(1.5,1) |
+------------+------------------+
|      0.5   |      -0.5        |
+------------+------------------+
```

ROUND

The `ROUND` function returns the rounded value of the `numeric` parameter.

Syntax

```
ROUND (numeric[,decimal])
```

Parameters

Parameter	Description
numeric	The expression of the numeric types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`.
decimal	If `decimal` is greater than or equal to 0, `numeric` is rounded to the number of decimal places specified by `decimal`. If `decimal` is less than 0, the numeric value is rounded to the specified number of digits to the left of the decimal point. The number of digits is specified by `decimal`. If `decimal` is not an integer, the integer part of `decimal` is truncated. If you do not specify `decimal`, `numeric` is rounded to the integer.

Return type

If you do not specify `decimal`, the return type is the same as the data type of the `numeric` parameter. If you specify `decimal`, the return type is the `NUMBER` data type.

Examples

This example shows the results of rounding 5555.6666 under different values of `decimal`.

Execute the following statement:

```
SELECT ROUND(5555.6666, 2.1), ROUND(5555.6666, -2.6), ROUND(5555.6666) FROM DUAL;
```

The following query result is returned:

```
+----------------------+-----------------------+------------------+
| ROUND(5555.6666,2.1) | ROUND(5555.6666,-2.6) | ROUND(5555.6666) |
+----------------------+-----------------------+------------------+
|              5555.67 |                  5600 |             5556 |
+----------------------+-----------------------+------------------+
```

SIGN

The `SIGN` function returns the sign of the number `n`. If n is greater than 0, 1 is returned. If n is smaller than 0, -1 is returned. If n is equal to 0, 0 is returned.

Syntax

```
SIGN (n)
```

Parameters

Parameter	Description
n	The expression of the exact numeric data types or the approximate numeric data types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`.

Return type

The numeric values 0, 1, and -1 are returned.

Examples

This example shows the results of using the `SIGN` function for three different numbers.

Execute the following statement:

```
SELECT SIGN(100), SIGN(-100), SIGN(0) FROM DUAL;
```

The following query result is returned:

```
+-----------+------------+---------+
| SIGN(100) | SIGN(-100) | SIGN(0) |
+-----------+------------+---------+
|         1 |         -1 |       0 |
+-----------+------------+---------+
```

SQRT

The `SQRT` function returns the square root of `n`.

Syntax

```
SQRT (n)
```

Parameters

Parameter	Description
n	The expression of the numeric types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`. `n` cannot be a negative number.

Return type

The return type is the same as the data type of the `n` parameter.

Examples

This example shows the results of using the `SQRT` function to calculate the square roots of two different numbers.

Execute the following statement:

```
SELECT SQRT(64), SQRT(10) FROM DUAL;
```

The following query result is returned:

```
+----------+-----------------------------------------+
| SQRT(64) | SQRT(10)                                |
+----------+-----------------------------------------+
|        8 | 3.16227766016837933199889354443271853372 |
+----------+-----------------------------------------+
```

TRUNC

The `TRUNC` function returns the `numeric` value that is truncated based on `precision`.

Syntax

```
TRUNC (numberic[,precision])
```

Parameters

Parameter	Description
numeric,precision	The expression of the numeric types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`. If `precision` is not an integer, `precision` is truncated to its integer part. If `precision` is greater than or equal to 0, `numeric` is truncated to the number of decimal places specified by `precision`. If `precision` is less than 0, `numeric` is truncated to `precision` digits to the left of the decimal point, and other data before the decimal point is represented by zero. The default value of `precision` is 0.

Return type

If `precision` is not specified, the return type is the same as the data type of the `numeric` parameter. If `precision` is specified, the return type is `NUMBER`.

Examples

This example shows the results of calculating 5555.66666 under different values of `precision`.

Execute the following statement:

```
SELECT TRUNC(5555.66666, 2.1), TRUNC(5555.66666, -2.6), TRUNC(5555.66666) FROM DUAL;
```

The following query result is returned:

```
+----------------------+-----------------------+--------------------+
| TRUNC(5555.66666,2.1) | TRUNC(5555.66666,-2.6) | TRUNC(5555.033333) |
+----------------------+-----------------------+--------------------+
|               5555.66 |                  5500 |               5555 |
+----------------------+-----------------------+--------------------+
```

17.1.5.6.2.2. String functions that return strings

CHR

The `CHR` function converts `n` to one or more equivalent characters. The return value is relevant to the character sets of the current system.

Syntax

```
CHR ( n )
```

Parameters

Parameter	Value range
n	0~4294967295

Return type

The return value is relevant to the character sets of the current system. The character sets that ApsaraDB for OceanBase supports are `UTF-8`, `UTF-16`, `GBK`, and `GB18030`.

Examples

Decimal (25700) > Hexadecimal (0x6464) > `UTF-8` encoding (dd)

Execute the following statement:

```
SELECT CHR(25700) AS str FROM DUAL;
```

The following query result is returned:

```
+------+
| STR  |
+------+
| dd   |
+------+
```

Decimal (50318) > Hexadecimal (0xC48E) > `UTF-8` encoding (Ď)

Execute the following statement:

```
SELECT CHR(50318) AS str FROM DUAL;
```

The following query result is returned:

```
+------+
| STR  |
+------+
| Ď    |
+------+
```

Decimal > `UTF-8` encoding

```
SELECT CHR(67)||CHR(65)||CHR(84) "Dog" FROM DUAL;
```

The following query result is returned:

```
+------+
| Dog  |
+------+
| CAT  |
+------+
```

CONCAT

The `CONCAT` function can concatenate two strings.

Syntax

```
CONCAT(c1,c2)
```

Parameters

Parameter	Description
c1	The string. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.
c1	The string. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.

Return type

The return type is the same as the data type of `c1`.

Examples

In this example, the `CONCAT` function is used to concatenate the `'010-'` character and the `'88888888'` character.

Execute the following statement:

```
SELECT concat('010-','88888888')||'to 23'XXXX phone FROM DUAL;
```

The following query result is returned:

```
+-------------------+
| XXXX phone        |
+-------------------+
| 010-88888888 to 23  |
+-------------------+
```

INITCAP

The `INITCAP` function returns a string where the initial letter of each word is in uppercase and the other letters are in lowercase.

Syntax

```
INITCAP(c1)
```

Parameters

Parameter	Description
c1	The string. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2`.

Return type

The data of the `CHAR` type is returned.

Examples

Execute the following statement:

```
SELECT initcap('smith abc aBC') upp FROM DUAL;
```

The following query result is returned:

```
+------------------+
|   UPP            |
+------------------+
|   Smith Abc Abc  |
+------------------+
```

LOWER

The `LOWER` function converts a string to lowercase.

Syntax

```
LOWER(c1)
```

Parameters

Parameter	Description
c1	The string. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.

Return type

The return type is the same as the data type of `c1`.

Examples

Execute the following statement:

```
SELECT  lower('AaBbCcDd')AaBbCcDd FROM DUAL;
```

The following query result is returned:

```
+-----------+
| AABBCCDD |
+-----------+
| aabbccdd |
+-----------+
```

LPAD

The `LPAD` function left-pads the `c1` string with the `c2` string until the length reaches `n`.

Syntax

```
LPAD(c1,n[,c2])
```

Parameters

Parameter	Description
c1	The string. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.
n	The total length of characters after the string is padded. The type must be the `NUMBER` integer type or the type that can be implicitly converted to the `NUMBER` integer type.
c2	The padding string. The default value is a space. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.

Return type

If `c1` is the data of the character type, the return type is `VARCHAR2`. If `c1` is of the national character data type, the return type is `NVARCHAR2`. If the data type of `c1` is `CLOB`, the return type is `CLOB`.

> **Notice**
>
> If the length of `c1` is greater than `n`, the function returns the leftmost `n` characters of `c1`. If the length of `c1` is less than `n` and the length of `c1` that is left-padded with `c2` is greater than `n`, the function returns the rightmost `n` characters of c1 that is left-padded with c2.

Examples

Execute the following statement:

```
SELECT  lpad('gao',10,'*') FROM DUAL;
```

The following query result is returned:

```
+--------------------+
| LPAD('GAO',10,'*') |
+--------------------+
| *******gao         |
+--------------------+
```

LTRIM

The `LTRIM` function removes the string that appears on the left.

Syntax

```
LTRIM(c1 [,c2])
```

The `LTRIM` function removes all the `c2` characters that are included in `c1` from the left end. If you do not specify `c2`, the default value is a single space.

Parameters

Parameter	Description
c1	The string. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.
c2	The string to be removed. The default value is a space. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.

Return type

If the data type of `c1` is `CHAR` or `VARCHAR2`, the function returns the `VARCHAR2` data type. If the data type of `c1` is `NCHAR` or `NVARCHAR2`, the function returns the `NVARCHAR2` data type. If the data type of `c1` is `CLOB`, the data type of the returned string is `CLOB`.

Examples

Execute the following statement:

```
SELECT LTRIM('   gao qian jing',' ') text FROM DUAL;
```

The following query result is returned:

```
+------------------+
|   TEXT           |
+------------------+
|   gao qian jing  |
+------------------+
```

REGEXP_REPLACE

The `REGEXP_REPLACE` function replaces regular expressions.

Syntax

```
REGEXP_REPLACE (source_char, pattern [,replace_string [, position [, occurrence [, match_param ]  ]
]  ])
```

Parameters

Parameter	Description
source_char	The character expression that is used as a search value. It is generally a character column and its data type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.
pattern	The regular expression. It is generally a text literal and its data type can be `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2`.
replace_string	The replacement character. The type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.
position	The positive integer type. It indicates the ordinal number of the character from which ApsaraDB for OceanBase starts to search the character of `source_char`. The default value is 1 and indicates that ApsaraDB for OceanBase starts to search `source_char` from the first character.

Parameter	Description
occurrence	The non-negative integer. It indicates that the replacement operation occurs. If you specify this parameter as 0, ApsaraDB for OceanBase replaces all the matching items. If you specify this parameter as a positive integer n, ApsaraDB for OceanBase replaces the matching items that occur for the nth time. By default, all the matching items are replaced. If you specify this parameter as 0, all the matching items are also replaced.
match_param	The character expression of the `VARCHAR2` or `CHAR` data type. It allows you to change the default matching behavior of the function. `i` indicates that the match is not case-sensitive. `c` indicates that the match is case-sensitive. `n` represents periods (.). The period (`.`) indicates that line feeds are not matched. `m` indicates the multi-line mode. `x` indicates that space characters are ignored. By default, space characters match with each other.

Return type

The returned result has the same data type as `source_char` .

Examples

The following example checks the string to search for two or more spaces. ApsaraDB for OceanBase replaces each occurrence of two or more spaces with one space. Execute the following statement:

```
SELECT REGEXP_REPLACE('500   OceanBase     Parkway,    Redwood  Shores, CA', '( ){2,}', ' ') "REGEXP_
REPLACE"
FROM DUAL;
```

The following query result is returned:

```
REGEXP_REPLACE
--------------------------------------
500 OceanBase Parkway, Redwood Shores, CA
```

REPLACE

The `REPLACE` function replaces some identical strings in the value of a character expression with new strings.

Syntax

```
REPLACE(c1,c2[,c3])
```

Parameters

Parameter	Description
c1	The CHAR string to be replaced.
c2	The string to be searched for and replaced.
c3	The replacement string. By default, this parameter is empty. This indicates the deletion instead of using spaces.

The data types of c1, c2, and c3 can be CHAR, VARCHAR2, NCHAR, NVARCHAR2, and CLOB.

Return type

The character set of the returned string is the same as that of c1. If c3 is the default or NULL, all the occurrences of c2 in c1 are removed. If c2 is NULL, the result is c1. If the data type of c1 is CLOB, the function returns the CLOB data type. If the data type of c1 is not CLOB, the function returns the VARCHA2 data type.

Examples

Execute the following statement:

```
SELECT replace('he love you','he','i') test FROM DUAL;
```

The following query result is returned:

```
+------------+
| TEST       |
+------------+
| i love you |
+------------+
```

RPAD

The RPAD function right-pads the c1 string with the c2 string until the length reaches n.

Syntax

```
RPAD(c1,n[,c2])
```

Parameters

Parameter	Description

Parameter	Description
c1	The string. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.
n	The total length of characters after the string is padded. The type must be the `NUMBER` integer type or the type that can be implicitly converted to the `NUMBER` integer type.
c2	The padding string. The default value is a space. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.

Return type

If `c1` is the data of the character type, the return type is `VARCHAR2`. If `c1` is of the national character data type, the return type is `NVARCHAR2`. If the data type of `c1` is `CLOB`, the return type is `CLOB`.

> **Note**
>
> - If the length of `c1` is greater than `n`, the function returns the leftmost `n` characters of `c1`.
> - If the length of `c1` is less than `n` and the length of `c1` that is right-padded with `c2` is greater than `n`, the function returns the leftmost `n` characters of c1 that is right-padded with c2.
> - If the length of `c1` is less than n and the length of the `c1` that is right-padded with `c2` is also less than `n`, the function returns the leftmost `n` characters of `c1` that is right-padded with multiple replicated `c2` (the total length of c1 after padding is greater than or equal to `n`).

Examples

Execute the following statement:

```
SELECT rpad('gao',10,'*a') FROM DUAL;
```

The following query result is returned:

```
+----------------------+
| RPAD('GAO',10,'*A') |
+----------------------+
| gao*a*a*a*           |
+----------------------+
```

RTRIM

The `RTRIM` function deletes a string that appears on the right side. This function is useful for formatting the output of a query.

Syntax

```
RTRIM(c1 [,c2])
```

`RTRIM` deletes `c2` from the right end of all the characters that appear in `c1`. If you do not specify `c2`, the default value is a single space.

Parameters

Parameter	Description
c1	The string.
c2	The string to be deleted. The default value is a space.

The data types of `c1` and `c2` can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, and `CLOB`.

Return type

If the data type of `c1` is `CHAR` or `VARCHAR2`, the function returns the `VARCHAR2` data type.

If the data type of `c1` is `NCHAR` or `NVARCHAR2`, the function returns the `NVARCHAR2` data type.

If the data type of `c1` is `CLOB`, the data type of the returned string is `CLOB`.

Examples

Execute the following statement:

```
SELECT RTRIM('gao qian jingXXXX','X') text FROM DUAL;
```

The following query result is returned:

```
+------------------+
|   TEXT           |
+------------------+
|   gao qian jing  |
+------------------+
```

SUBSTR

The `SUBSTR` function extracts a substring. A multibyte character, such as a Chinese character and a full-width character, is calculated as one character.

Syntax

```
SUBSTR(c1,n1[,n2])
```

Parameters

Parameter	Description
c1	The string to be truncated. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.
n1	The start position for truncating the string. If n1 is equal to 0 or 1, the string is truncated from the first character.
n2	The length of the truncated string. If you do not specify `n2`, the truncated string starts from the `n1` th character to the end.

Return type

The data that has the same type as `c1` is returned. If `n2` is 0, it is considered as 1. If `n2` is positive, ApsaraDB for OceanBase searches for the first character from the beginning of `c1`. If `n2` is negative, ApsaraDB for OceanBase counts backward `c1` from the end of `c1`. If `n3` is omitted, ApsaraDB for OceanBase returns all the characters of `c1`. If `n3` is less than 1, ApsaraDB for OceanBase returns `NULL`.

Examples

Execute the following statement:

```
SELECT SUBSTR('13088888888',3,8) test FROM DUAL;
```

The following query result is returned:

```
+----------+
| TEST     |
+----------+
| 08888888 |
+----------+
```

TRANSLATE

The `TRANSLATE` function replaces specified characters in the value of a character expression with new characters. A multibyte character, such as a Chinese character and a full-width character, is calculated as one character.

Syntax

```
TRANSLATE(c1,c2,c3)
```

Parameters

Parameter	Description
c1	The character or the variable that you want to replace.
c2	The original character set to be queried.
c3	The new character set for replacement. Characters in a sequence in `c2` are replaced with characters in the corresponding sequence in `c3`.

Note

- The data types of `c1`, `c2`, and `c3` can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.
- If the length of `c3` is greater than that of `c2`, the extra characters in `c3` are invalid.
- If the length of `c3` is less than that of `c2`, extra characters in `c2` are replaced with null. This indicates that extra characters are deleted.
- If the length of `c3` is 0, an empty string is returned.
- If `c2` contains duplicate characters, the character in the position of the first occurrence is replaced.

Return type

The characters of the `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB` type are returned.

Examples

Execute the following statement:

```
SELECT TRANSLATE('he love you','he','i') FROM DUAL;
```

The following query result is returned:

```
+------------------+
|   TRANSLATE      |
+------------------+
|   i love you     |
+------------------+
```

TRIM

The `TRIM` function deletes leading or trailing characters (or both) from a string.

Syntax

```
TRIM([ { { LEADING | TRAILING | BOTH } [ trim_character ] | trim_character } FROM ] trim_source)
```

Parameters

Parameter	Description
LEADING	The leading characters.
TRAILING	The trailing characters.
BOTH	The leading and trailing characters.
trim_character	The characters to be deleted.
trim_source	The trim source.

The data type of `trim_char` and `trim_source` can be `VARCHAR2` or data types that can be implicitly converted to `VARCHAR2`. If you specify `LEADING`, ApsaraDB for OceanBase deletes all the leading characters that are equal to `trim_character`. If you specify `TRAILING`, ApsaraDB for OceanBase deletes all the trailing characters that are equal to `trim_character`. If you specify `BOTH` or none of the three parameters, ApsaraDB for OceanBase deletes the leading and trailing characters that are equal to `trim_character`. If you do not specify `trim_character`, the default value is a blank space. If you specify only `trim_source`, ApsaraDB for OceanBase deletes the leading and trailing spaces. If the value that is returned by the function is of the `VARCHAR2` data type, the maximum length of the value is `trim_source`.

Return type

If the data type of `trim_source` is `CHAR` or `VARCHAR2`, the function returns the `VARCHAR2` data type.

If the data type of `trim_source` is `NCHAR` or `NVARCHAR2`, the function returns the `NVARCHAR2` data type.

If the data type of `trim_source` is `CLOB`, the function returns the `CLOB` data type.

If `trim_source` or `trim_character` is `NULL`, the `TRIM` function returns `NULL`.

Examples

Execute the following statement:

```
SELECT TRIM('X' from 'XXXgao qian jingXXXX'),TRIM('X' from 'XXXgaoXXjingXXXX') text FROM DUAL;
```

The following query result is returned:

```
+-----------------------------------+-----------+
| TRIM('X'FROM'XXXGAOQIANJINGXXXX') | TEXT      |
+-----------------------------------+-----------+
| gao qian jing                     | gaoXXjing |
+-----------------------------------+-----------+
```

UPPER

The `UPPER` function converts a string to uppercase.

Syntax

```
UPPER(c1)
```

Parameters

Parameter	Description
c1	The string. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.

Return type

The characters of the `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB` type are returned.

Examples

Execute the following statement:

```
SELECT UPPER('AaBbCcDd') upper FROM DUAL;
```

The following query result is returned:

```
+----------+
| UPPER    |
+----------+
| AABBCCDD |
+----------+
```

17.1.5.6.2.3. String functions that return numbers

ASCII

The `ASCII` function returns the ASCII code value of the leftmost character of a character expression.

Syntax

```
ASCII(x)
```

Parameters

Parameter	Description
x	The expression of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type.

Return type

`NUMBER` data type

Examples

In this example, the `ASCII` function returns the ASCII code values of **A**, **a**, **Medium**, and spaces.

```
SELECT ASCII('A') A, ASCII('a') a, ASCII(' ') space,ASCII('Medium') hz FROM DUAL;
```

The following result is returned:

```
+----+----+-------+-----+
| A  | a  | space |  hz |
+----+----+-------+-----+
| 65 | 97 |    32 | 228 |
+----+----+-------+-----+
```

INSTR

The `INSTR` function searches for a specified character in a string and returns the position of the specified character.

> **Notice**
> A multibyte character, such as a Chinese character and a full-width character, is calculated as one character.

Syntax

```
INSTR(c1,c2[,i[,j]])
```

Parameters

Parameter	Description
c1	The string to be searched. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.

Parameter	Description
c2	The string that you want to search for. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.
i	The position where the search starts. Default value: 1. If the value is less than 0, the search starts from the opposite direction, but the function returns the left-to-right position of the searched characters.
j	The position of the `j` th occurrence. Default value: 1.

Return type

`NUMBER` data type

Examples

In this example, the function needs to return the position of the second ce occurrence for **instring1** when the function searches forward for **ce**. The function needs to return the position of the second ce occurrence for **instring2** when the function searches backward for **ce**.

```
SELECT INSTR ('oceanbase pratice','ce',1,2) instring1,INSTR ('oceanbase pratice','ce',-1,2) instring2
FROM DUAL;
```

The result is returned. For the forward search, the second occurrence of **ce** lies in the sixteenth character. For the backward search, the second occurrence of **ce** lies in the second character.

```
+----------+------------+
| instring1 | instring2 |
+----------+------------+
|       16 |          2 |
+----------+------------+
```

LENGTH

The `LENGTH` function returns the length of a string.

> **Notice**
> A multibyte character, such as a Chinese character and a full-width character, is calculated as one character.

Syntax

```
LENGTH(c1)
```

Parameters

Parameter	Description
c1	The string of the `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB` data type.

Return type

`NUMBER` data type

Examples

This example shows the lengths of the strings: **Test**, **Haidian District of Beijing**, and **Beijing TO_CHAR**.

```
SELECT LENGTH ('Test'), LENGTH('Haidian District of Beijing'), LENGTH('Beijing TO_CHAR') FROM DUAL;
```

The following result is returned:

```
+------------------+------------------------------+-------------------------+
|    LENGTH('Test')|         LENGTH('Haidian District of Beijing') |   LENGTH('Bejing TO_CHAR') |
+------------------+------------------------------+-------------------------+
|                2 |                            6 |                       9 |
+------------------+------------------------------+-------------------------+
```

17.1.5.6.2.4. Datetime functions

ADD_MONTHS

The `ADD_MONTHS` function returns a date value that is `n` months after `date`. If `n` is a negative value, this function returns a date value that is `n` months before `date`.

> **Notice**
>
> Different months have different numbers of days. If `date` is the last day of a month, this function returns the last day of the resulting month. For example, if you use `ADD_MONTHS` to calculate the date one month before March 31, 2020, the function returns February 29, 2020.

Syntax

```
ADD_MONTHS (date,n)
```

Parameters

Parameter	Description
date	A value of the `DATE` data type.
n	A value of the `NUMBER` data type.

Return type

The return type is `DATE`.

Examples

Example 1: In the following example, the date that is three months after the system date is queried.

Execute the following statement:

```
SELECT SYSDATE, ADD_MONTHS(SYSDATE,3) FROM DUAL;
```

The following result is returned:

```
+---------------------+-----------------------+
| SYSDATE             | ADD_MONTHS(SYSDATE,3) |
+---------------------+-----------------------+
| 2020-03-26 12:21:40 | 2020-06-26 12:21:40   |
+---------------------+-----------------------+
```

Example 2: In the following example, the date that is three months before the system date is queried.

Execute the following statement:

```
SELECT SYSDATE, ADD_MONTHS(SYSDATE,-3) FROM DUAL;
```

The following result is returned:

```
+---------------------+-----------------------+
| SYSDATE             | ADD_MONTHS(SYSDATE,-3) |
+---------------------+-----------------------+
| 2020-03-26 12:21:04 | 2019-12-26 12:21:04    |
+---------------------+-----------------------+
```

CURRENT_DATE

The `CURRENT_DATE` function returns the current date in the session time zone.

Syntax

```
CURRENT_DATE
```

Parameters

No parameters are involved.

Return type

The return type is `DATE`.

Examples

In the following examples, the `CURRENT_DATE` function returns the current date in different session time zones.

Set the current time zone to the UTC-5 time zone:

```
ALTER SESSION SET TIME_ZONE = '-05:00';
```

Execute the following statement to call the function:

```
SELECT CURRENT_DATE FROM DUAL;
```

The following result is returned:

```
+---------------------+
| CURRENT_DATE        |
+---------------------+
| 2020-03-08 01:40:11 |
+---------------------+
```

Change the current time zone to the UTC+8 time zone:

```
ALTER SESSION SET TIME_ZONE = '+08:00';
```

Execute the following statement to call the function:

```
SELECT CURRENT_DATE FROM DUAL;
```

The following result is returned:

```
+---------------------+
| CURRENT_DATE        |
+---------------------+
| 2020-03-08 14:40:11 |
+---------------------+
```

CURRENT_TIMESTAMP

The `CURRENT_TIMESTAMP` function returns the current date in the time zone of the current session. The return value is of the `TIMESTAMP WITH TIME ZONE` data type and contains the information about the current time zone.

Syntax

```
CURRENT_TIMESTAMP (precision)
```

Parameters

Parameter	Description

precision	The precision of the fractional part of seconds. Default value: 6. Valid values: 0 to 9.

Return type

`TIMESTAMP WITH TIME ZONE` data type that includes the information about the current time zone in its value

Examples

In the following examples, the `CURRENT_TIMESTAMP` function returns different results for different time zones of sessions.

Set the current time zone to GMT-5:

```
ALTER SESSION SET TIME_ZONE = '-05:00';
```

Execute the following statement to call the function:

```
SELECT CURRENT_TIMESTAMP FROM DUAL;
```

The following query result is returned:

```
+-----------------------------------+
| CURRENT_TIMESTAMP                 |
+-----------------------------------+
| 2020-03-08 01:49:31.219066 -05:00  |
+-----------------------------------+
```

Change the current time zone to GMT+8 and change the precision of the fractional part of seconds to 3:

```
ALTER SESSION SET TIME_ZONE = '+08:00';
```

Execute the following statement to call the function:

```
SELECT CURRENT_TIMESTAMP(3) FROM DUAL;
```

The following query result is returned:

```
+---------------------------------+
| CURRENT_TIMESTAMP(3)            |
+---------------------------------+
| 2020-03-08 14:50:32.499 +08:00  |
+---------------------------------+
```

DBTIMEZONE

The `DBTIMEZONE` function returns the time zone of the current database instance. In ApsaraDB for OceanBase, the time zone is always +00:00 and cannot be changed.

Syntax

```
DBTIMEZONE
```

Parameters

None

Return type

`VARCHAR2` data type

Examples

Execute the following statement:

```
SELECT DBTIMEZONE FROM DUAL;
```

The following query result is returned:

```
+------------+
| DBTIMEZONE |
+------------+
| +00:00     |
+------------+
```

EXTRACT(datetime)

The `EXTRACT(datetime)` function extracts elements such as the year, month, day, hour, minute, and second values from a specified time field or expression.

Syntax

```
EXTRACT  (fields FROM datetime)
```

Parameters

Parameter	Description
fields	The name of the element to be extracted. Valid values: `YEAR`, `MONTH`, `DAY`, `HOUR`, `MINUTE`, `SECOND`, `TIMEZONE_HOUR`, `TIMEZONE_MINUTE`, `TIMEZONE_REGION`, and `TIMEZONE_ABBR`.
datetime	A value of the data types such as `DATE`, `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE`, `TIMESTAMP WITH LOCAL TIME ZONE`, `INTERVAL YEAR TO MONTH`, and `INTERVAL DAY TO SECOND`.

Return type

When this function extracts the `TIMEZONE_REGION` and `TIMEZONE_ABBR` elements, the return type is `VARCHAR2`. When this function extracts other elements, the return type is `NUMBER`.

Examples

Execute the following statement:

```
SELECT EXTRACT(HOUR FROM TIMESTAMP '2001-2-16 2:38:40') Hour,
        EXTRACT(MINUTE FROM TIMESTAMP '2001-2-16 2:38:40 ') Minute,
            EXTRACT(SECOND FROM TIMESTAMP '2001-2-16 2:38:40 ') Second,
            EXTRACT(DAY FROM TIMESTAMP '2001-2-16 2:38:40 ') Day,
            EXTRACT(MONTH FROM TIMESTAMP '2001-2-16 2:38:40 ') Month,
            EXTRACT(YEAR FROM TIMESTAMP '2001-2-16 2:38:40 ') Year
FROM DUAL;
```

The following result is returned:

```
+--------+--------+------+------+------+------+
| Hour   | Minute   | Second   | Day   | Month   | Year   |
+--------+--------+------+------+------+------+
|      2 |     38 |   40 |   16 |    2 | 2001 |
+--------+--------+------+------+------+------+
```

FROM_TZ

The `FROM_TZ` function combines a `TIMSTAMP` value with the time zone information into a time value of the `TIMESTAMP WITH TIME ZONE` data type.

Syntax

```
FROM_TZ (timestamp_value,time_zone_value)
```

Parameters

Parameter	Description
timestamp_value	A time value of the `TIMSTAMP` data type.
time_zone_value	The time zone information.

Return type

`TIMESTAMP WITH TIME ZONE` data type

Examples

Execute the following statement:

```
SELECT FROM_TZ(TIMESTAMP '2020-03-28 08:00:00', '-03:00') FROM DUAL;
```

The following query result is returned:

```
+-----------------------------------------------+
| FROM_TZ(TIMESTAMP'2020-03-2808:00:00','-03:00') |
+-----------------------------------------------+
| 2020-03-28 08:00:00.000000000 -03:00          |
+-----------------------------------------------+
```

LAST_DAY

The `LAST_DAY` function returns the date of the last day of the month that contains `date`.

Syntax

```
LAST_DAY (date)
```

Parameters

Parameter	Description
date	A value of a data type that contains the date information, such as `DATE`, `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE`, and `TIMESTAMP WITH LOCAL TIME ZONE`.

Return type

`DATE` data type

Examples

In this example, the following information is queried: the current date, the last day of the current month, and the number of the remaining days in the current month.

Execute the following statement:

```
SELECT SYSDATE, LAST_DAY(SYSDATE)Last, LAST_DAY(SYSDATE)-SYSDATE "Left" FROM DUAL;
```

The following query result is returned:

```
+---------------------+---------------------+------+
| SYSDATE             | LAST                | Left |
+---------------------+---------------------+------+
| 2020-03-08 15:23:33 | 2020-03-31 15:23:33 |   23 |
+---------------------+---------------------+------+
```

LOCALTIMESTAMP

The `LOCALTIMESTAMP` function returns the current date in the time zone of the current session. The return type is `TIMESTAMP`. This function and the CURRENT_TIMESTAMP function return different data types. The `CURRENT_TIMESTAMP` function returns a value of the `TIMESTAMP WITH TIME ZONE` data type.

Syntax

```
LOCALTIMESTAMP (timestamp_precision)
```

Parameters

Parameter	Description
timestamp_precision	The precision of fractional seconds. The value ranges from 0 to 9, and the default value is 6.

Return type

`TIMESTAMP` data type

Examples

The following examples show the results that the `LOCALTIMESTAMP` function returns in different session time zones.

Set the current time zone to the UTC-5 time zone:

```
ALTER SESSION SET TIME_ZONE = '-05:00';
```

Execute the following statement to call the function:

```
SELECT LOCALTIMESTAMP FROM DUAL;
```

The following query result is returned:

```
+----------------------------+
| LOCALTIMESTAMP             |
+----------------------------+
| 2020-03-08 02:30:20.062104 |
+----------------------------+
```

Change the current time zone to the UTC+8 time zone and change the precision of fractional seconds to 3:

```
ALTER SESSION SET TIME_ZONE = '+08:00';
```

Execute the following statement to call the function:

```
SELECT LOCALTIMESTAMP(3) FROM DUAL;
```

The following query result is returned:

```
+-------------------------+
| LOCALTIMESTAMP(3)       |
+-------------------------+
| 2020-03-08 15:30:54.500 |
+-------------------------+
```

MONTHS_BETWEEN

The `MONTHS_BETWEEN` function returns the number of months between `date1` and `date2`.

Syntax

```
MONTHS_BETWEEN(date1,date2)
```

Parameters

Parameter	Description
date1	A value of the `DATE` data type.
date2	A value of the `DATE` data type.

Return type

The return type is `NUMBER`. If `date1` is greater than `date2`, this function returns a positive number. If `date1` is less than `date2`, this function returns a negative number.

Examples

The following statement queries the number of months between the current time and a specified time:

```
SELECT SYSDATE,
MONTHS_BETWEEN(SYSDATE,TO_DATE('2006-01-01','YYYY-MM-DD')),
MONTHS_BETWEEN(SYSDATE,TO_DATE('2022-01-01','YYYY-MM-DD'))
FROM DUAL;
```

The following result is returned:

```
+---------------------+-----------------------------------------------------------------+-----------------
-----------------------------------------+
| SYSDATE             | MONTHS_BETWEEN(SYSDATE,TO_DATE('2006-01-01','YYYY-MM-DD')) | MONTHS_BETWEEN(S
YSDATE,TO_DATE('2022-01-01','YYYY-MM-DD')) |
+---------------------+-----------------------------------------------------------------+-----------------
-----------------------------------------+
| 2020-03-08 15:38:35 |                   170.24683206391875746714456391875746714 5 |
-21.753167936081242532855436081242532855544 |
+---------------------+-----------------------------------------------------------------+-----------------
-----------------------------------------+
```

NEXT_DAY

The `NEXT_DAY` function returns the date value of the weekday `c1` in the week following the date `d1`.

Syntax

```
NEXT_DAY (d1[,c1])
```

Parameters

Parameter	Description
d1	A value of the `DATE` data type.
c1	A weekday value. Valid values: `MONDAY`, `TUESDAY`, `WEDNESDAY`, `THURSDAY`, `FRIDAY`, `SATURDAY`, and `SUNDAY`.

Return type

`DATE` data type

Examples

The following statements are used to query the date values that correspond to each weekday value in the next week:

```
SELECT SYSDATE Current date,
NEXT_DAY(SYSDATE,'MONDAY') Next Monday,
NEXT_DAY(SYSDATE,'TUESDAY') Next Tuesday,
NEXT_DAY(SYSDATE,'WEDNESDAY') Next Wednesday,
NEXT_DAY(SYSDATE,'THURSDAY') Next Thursday,
NEXT_DAY(SYSDATE,'FRIDAY') Next Friday,
NEXT_DAY(SYSDATE, 'Saturn ') Next Saturday,
NEXT_DAY(SYSDATE,'SUNDAY') Next Sunday
FROM DUAL;
```

The following query result is returned:

```
+---------------------+---------------------+---------------------+---------------------+-----------
---------+---------------------+---------------------+---------------------+
| Current date            | Next Monday           | Next Tuesday          | Next Wednesday          |
Next Thursday           | Next Friday          | Next Saturday           | Next Sunday          |
+---------------------+---------------------+---------------------+---------------------+-----------
---------+---------------------+---------------------+---------------------+
| 2020-03-08 15:47:57 | 2020-03-09 15:47:57 | 2020-03-10 15:47:57 | 2020-03-11 15:47:57 | 2020-03-12
15:47:57 | 2020-03-13 15:47:57 | 2020-03-14 15:47:57 | 2020-03-15 15:47:57 |
+---------------------+---------------------+---------------------+---------------------+-----------
---------+---------------------+---------------------+---------------------+
```

NUMTODSINTERVAL

The `NUMTODSINTERVAL` function converts the `n` parameter value to a value of the `INTERVAL DAY TO SECOND` data type. The `interval_unit` parameter specifies the unit of the result value.

Syntax

```
NUMTODSINTERVAL (n,interval_unit)
```

Parameters

Parameter	Description
n	The value of the `NUMBER` data type.
interval_unit	Valid values for the unit: `DAY`, `HOUR`, `MINUTE`, and `MINUTE`.

Return type

`INTERVAL DAY TO SECOND` data type

Examples

In the following example, the statement returns a datetime value and the value indicates the date and time that is 3 hours after the current date:

```
SELECT SYSDATE,SYSDATE+NUMTODSINTERVAL(3,'HOUR') AS RES FROM DUAL;
```

The following query result is returned:

```
+---------------------+---------------------+
| SYSDATE             | RES                 |
+---------------------+---------------------+
| 2020-03-08 16:01:40 | 2020-03-08 19:01:40 |
+---------------------+---------------------+
```

NUMTOYMINTERVAL

The `NUMTOYMINTERVAL` function converts the `n` parameter value to a value of the `INTERVAL YEAR TO MONTH` data type in the unit specified by `interval_unit`.

Syntax

```
NUMTOYMINTERVAL (n,interval_unit)
```

Parameters

Parameter	Description
n	A value of the `NUMBER` data type.
interval_unit	The unit. Valid values: `YEAR` and `MONTH`.

Return type

The return type is `INTERVAL YEAR TO MONTH`.

Examples

In the following example, a date and time value that is three years after the current date is returned:

```
SELECT SYSDATE,SYSDATE+NUMTOYMINTERVAL(3,'YEAR') AS RES FROM DUAL;
```

The following result is returned:

```
+---------------------+---------------------+
| SYSDATE             | RES                 |
+---------------------+---------------------+
| 2020-03-08 16:03:58 | 2023-03-08 16:03:58 |
+---------------------+---------------------+
```

ROUND (date)

The `ROUND(date)` function returns a date and time value that is nearest to the specified `date`. The `fmt` parameter specifies the unit in which the interval between the return value and the specified date is measured.

Syntax

```
ROUND(date,[fmt])
```

Parameters

Parameter	Description
date	A value of the following data types: `DATE`, `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE`, and `TIMESTAMP WITH LOCAL TIME ZONE`. The values of all these data types include date values.
fmt	The unit in which the interval between the return value and the specified `date` is measured. The following table lists the valid values of this parameter, which are not case-sensitive.

Values of the fmt parameter	Description
j	The date and time value that corresponds to the nearest 00:00:00 is returned. This is the default value.
day, dy, and d	The nearest Sunday of the specified date is returned.

Values of the fmt parameter	Description
month, mon, mm, and rm	The first date of the month nearest to the specified date is returned.
q	The first date of the quarter nearest to the specified date is returned.
syear, year, yyyy, yyy, yy, and y	The first date of the year nearest to the specified date. The number of the y letters determines the precision of the interval value.
cc and scc	The first date of the century nearest to the specified date.

Return type

The return type is `DATE`.

Examples

Execute the following statement:

```
SELECT SYSDATE Current date,
ROUND(SYSDATE) The date and time value of the nearest 00:00:00,
ROUND(SYSDATE,'DAY') The nearest Sunday,
ROUND(SYSDATE,'MONTH') The first date of the nearest month,
ROUND(SYSDATE,'Q') The first date of the nearest quarter,
ROUND(SYSDATE,'YEAR') The first date of the nearest year
FROM DUAL;
```

The following result is returned:

```
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
| Current date            | The date and time value of the nearest 00:00:00        | The nearest Sunday          | The first date of the nearest month            | The first date of the nearest quarter | The first date of the nearest year        |
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
| 2020-03-08 20:24:53 | 2020-03-09 00:00:00 | 2020-03-08 00:00:00 | 2020-03-01 00:00:00 | 2020-04-01 00:00:00 | 2020-01-01 00:00:00 |
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
```

SESSIONTIMEZONE

The `SESSIONTIMEZONE` function returns the time zone of the current session.

Syntax

```
SESSIONTIMEZONE
```

Parameters

None

Return type

`VARCHAR2` data type

Examples

The following statement is used to query the database time zone and the time zone of the current session:

```
SELECT DBTIMEZONE,SESSIONTIMEZONE FROM DUAL;
```

The following query result is returned:

```
+------------+-----------------+
| DBTIMEZONE | SESSIONTIMEZONE |
+------------+-----------------+
| +00:00     | +08:00          |
+------------+-----------------+
```

You can change the time zone of the current session by using the `ALTER SESSION` statement, but you cannot change the database time zone.

```
ALTER SESSION SET TIME_ZONE = '+05:00';
```

Execute the following statement to query the time zone of the current session after the change:

```
SELECT DBTIMEZONE,SESSIONTIMEZONE FROM DUAL;
```

The following query result is returned:

```
+------------+-----------------+
| DBTIMEZONE | SESSIONTIMEZONE |
+------------+-----------------+
| +00:00     | +05:00          |
+------------+-----------------+
```

SYS_EXTRACT_UTC

The `SYS_EXTRACT_UTC` function returns the standard Coordinated Universal Time (UTC) time that corresponds to the specified time.

Syntax

```
SYS_EXTRACT_UTC (datetime_with_timezone)
```

> **Notice**
>
> UTC is short for Coordinated Universal Time. The UTC and Greenwich Mean Time (GMT) are the same as the local time in London, UK.

Parameters

Parameter	Description
datetime_with_timezone	The values of the `TIMESTAMP WITH TIME ZONE` and `TIMESTAMP WITH LOCAL TIME ZONE` data types.

Return type

`TIMESTAMP` data type

Examples

Execute the following statement:

```
SELECT SYS_EXTRACT_UTC(TIMESTAMP '2020-03-28 11:30:00.00 -08:00')
FROM DUAL;
```

The following query result is returned:

```
+-----------------------------------------------------------+
| SYS_EXTRACT_UTC(TIMESTAMP'2020-03-2811:30:00.00-08:00') |
+-----------------------------------------------------------+
| 2020-03-28 19:30:00.000000000                             |
+-----------------------------------------------------------+
```

SYSDATE

The `SYSDATE` function returns the current date and time.

> **Notice**
>
> This function is not dependent on the time zone of the current session but on the time zone of the Linux operating system that runs on the database host.

Syntax

```
SYSDATE
```

Parameters

None

Return type

`DATE` data type

Examples

In the following example, the current time is returned in the specified format:

```
SELECT TO_CHAR
(SYSDATE, 'MM-DD-YYYY HH24:MI:SS') "NOW"
FROM DUAL;
```

The following query result is returned:

```
+---------------------+
| NOW                 |
+---------------------+
| 03-08-2020 20:44:04 |
+---------------------+
```

SYSTIMESTAMP

The `SYSTIMESTAMP` function returns the current date of the system. The return value contains the information about the current time zone. The fractional part of seconds in the return value consists of six digits and the six digits specify the precision.

> **Notice**
>
> This function is not dependent on the time zone of the current session but on the time zone of the Linux operating system that runs on the database host.

Syntax

```
SYSTIMESTAMP
```

Parameter

None

Return type

`TIMESTAMP WITH TIME ZONE` data type

Examples

Execute the following statement:

```
SELECT SYSTIMESTAMP FROM DUAL;
```

The following query result is returned:

```
+-----------------------------------+
| SYSTIMESTAMP                      |
+-----------------------------------+
| 2020-03-08 20:47:08.254086 +08:00  |
+-----------------------------------+
```

TO_CHAR (datetime)

The `TO_CHAR` function converts a value of the data types such as `DATE`, `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE`, `TIMESTAMP WITH LOCAL TIME ZONE`, `INTERVAL DAY TO SECOND`, and `INTERVAL YEAR TO MONTH` to a value of the `VARCHAR2` data type in the format specified by the `fmt` parameter. If you do not specify the `fmt` parameter, this function converts the `datetime` parameter value to a value of the `VARCHAR2` data type in the following format:

- The values of the `DATE`, `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE`, and `TIMESTAMP WITH LOCAL TIME ZONE` data types are converted to the default format of date and time values in the database. You can view the default format for values of each date and time data type in the "Data types" section.
- The values of the `INTERVAL DAY TO SECOND` and `INTERVAL YEAR TO MONTH` data types are converted to interval values in the numeric format.

Syntax

```
TO_CHAR({ datetime | interval } [, fmt [, 'nlsparam' ] ])
```

Parameters

Parameter	Description
datetime	A value of the data types such as `DATE`, `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE`, `TIMESTAMP WITH LOCAL TIME ZONE`, `INTERVAL DAY TO SECOND`, and `INTERVAL YEAR TO MONTH`.
fmt	The output format.
nlsparam	The language in which the month and day values are returned.

Return type

The return type is `VARCHAR2`.

Examples

Example 1: The following statement uses the TO_CHAR function to return the current system date and convert the date and time value to a result in the **DS DL** format:

```
SELECT TO_CHAR(SYSDATE,'DS DL') FROM DUAL;
```

The following result is returned:

```
+-----------------------------------+
| TO_CHAR(SYSDATE,'DSDL')           |
+-----------------------------------+
| 03/08/2020 Sunday, March 08, 2020 |
+-----------------------------------+
```

Example 2: The following statement converts an interval value to a result in a specified format and sets the return language to **AMERICAN**:

```
SELECT TO_CHAR(interval'1' year, 'SS-MI-HH', 'nls_language = AMERICAN') FROM DUAL;
```

The following result is returned:

```
+------------------------------------------------------------+
| TO_CHAR(INTERVAL'1'YEAR,'SS-MI-HH','NLS_LANGUAGE=AMERICAN') |
+------------------------------------------------------------+
| +01-00                                                     |
+------------------------------------------------------------+
```

TO_DSINTERVAL

The `TO_DSINTERVAL` function converts a string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type to a value of the `INTERVAL DAY TO SECOND` data type. You can add and subtract a date and time value by using this function.

Syntax

```
TO_DSINTERVAL(days hours:minutes:seconds[.frac_secs])
```

Parameters

Parameter	Description
days hours:minutes:seconds[.frac_secs]	A string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type in the format specified by this parameter.

Return type

`INTERVAL DAY TO SECOND` data type

Examples

The following example returns the date and time value that is 100 days after the current time:

```
SELECT SYSDATE, SYSDATE+TO_DSINTERVAL('100 00:00:00') FROM DUAL;
```

The following query result is returned:

```
+---------------------+---------------------------------------+
| SYSDATE             | SYSDATE+TO_DSINTERVAL('10000:00:00') |
+---------------------+---------------------------------------+
| 2020-03-26 12:40:39 | 2020-07-04 12:40:39                   |
+---------------------+---------------------------------------+
```

TO_TIMESTAMP

The `TO_TIMESTAMP` function converts a string to a value of the `TIMESTAMP` data type.

Syntax

```
TO_TIMESTAMP(char,[fmt],['nlsparam'])
```

Parameters

	Description
char	A string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type.
fmt	The format of the return value.
nlsparam	The language that is used to return the month and day values.

Return type

`TIMESTAMP` data type

Examples

Execute the following statement:

```
SELECT TO_TIMESTAMP ('10-Sep-02 14:10:10.123000', 'DD-Mon-RR HH24:MI:SS.FF')
FROM DUAL;
```

The following query result is returned:

```
+---------------------------------------------------------------------+
| TO_TIMESTAMP('10-SEP-0214:10:10.123000','DD-MON-RRHH24:MI:SS.FF') |
+---------------------------------------------------------------------+
| 2002-09-10 14:10:10.123000000                                       |
+---------------------------------------------------------------------+
```

TO_TIMESTAMP_TZ

The `TO_TIMESTAMP_TZ` function converts a string to a value that is of the `TIMESTAMP WITH TIME ZONE` data type and contains the time zone information.

Syntax

```
TO_TIMESTAMP_TZ (char,[fmt],['nlsparam'])
```

Parameters

Parameter	Description

char	Specifies the string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type.
fmt	Specifies the output format.
nlsparam	Specifies the language that is used for the returned month and day.

Return type

`TIMESTAMP WITH TIME ZONE` data type

Examples

Execute the following statement:

```
SELECT TO_TIMESTAMP_TZ ('10-Sep-02 14:10:10.123000', 'DD-Mon-RR HH24:MI:SS.FF') FROM DUAL;
```

The following query result is returned:

```
+---------------------------------------------------------------------------+
| TO_TIMESTAMP_TZ('10-SEP-0214:10:10.123000','DD-MON-RRHH24:MI:SS.FF') |
+---------------------------------------------------------------------------+
| 2002-09-10 14:10:10.123000000 +08:00                                      |
+---------------------------------------------------------------------------+
```

TO_YMINTERVAL

The `TO_YMINTERVAL` function converts a string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type to a value of the `INTERVAL YEAR TO MONTH` data type. You can use this function to add a time interval to or subtract a time interval from a datetime value.

Syntax

```
TO_YMINTERVAL (years-months)
```

Parameters

Parameter	Description
years-months	The string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type. The string must meet the format requirements of this parameter.

Return type

`INTERVAL YEAR TO MONTH` data type

Examples

In the following example, the statement returns a datetime value and the value indicates the date and time that is one year and two months later than the current time:

```
SELECT SYSDATE,SYSDATE+TO_YMINTERVAL('01-02') FROM DUAL;
```

The following query result is returned:

```
+---------------------+---------------------------------+
| SYSDATE             | SYSDATE+TO_YMINTERVAL('01-02') |
+---------------------+---------------------------------+
| 2020-03-08 22:32:01 | 2021-05-08 22:32:01            |
+---------------------+---------------------------------+
```

TRUNC (date)

The `TRUNC` function truncates an input value based on the specified unit. This function returns a datetime value that indicates the nearest date and time to the specified `date` based on the specified interval requirement. The `fmt` parameter specifies the unit for the time interval between the return value and the specified date. The returned date occurs before the date that is specified by `date`.

> **Notice**
>
> `TRUNC` is different from `ROUND` in the following aspect: The TRUNC function returns the nearest date to the specified `date` and the nearest date occurs before the specified `date`. `ROUND` returns the nearest date to the specified `date` and the nearest date occurs before or after the specified `date`.

Syntax

```
TRUNC (date,[fmt])
```

Parameters

Parameter	Description
date	The `DATE` data type.
fmt	The unit for the time interval between the return value of the function and the specified `date`. The following table lists the valid values of this parameter. The values of this parameter are not case-sensitive.

fmt parameter	Description
j	The default value. The function returns the nearest date to 00:00.
day, dy, or d	Returns the nearest Sunday to the specified date.
month, mon, mm, or rm	Returns the first date of the nearest month to the specified date.
q	Returns the first date of the nearest quarter to the specified date.
yyyy, yyy, yy, or y	Returns the first date of the nearest year to the specified date. yyyy, yyy, yy, and y correspond to different precisions.
cc or scc	Returns the first date of the nearest century to the specified date.

Return type

`DATE` data type

Examples

In the following example, the `TRUNC` function calculates the nearest dates to the **SYSDATE** based on the specified requirements:

```
SELECT SYSDATE The current date,
TRUNC(SYSDATE) The date of today,
TRUNC(SYSDATE,'DAY') The Sunday of this week,
TRUNC(SYSDATE,'MONTH') The first date of this month,
TRUNC(SYSDATE,'Q') The first date of this quarter,
TRUNC(SYSDATE,'YEAR') The first date of this year FROM DUAL;
```

The following query result is returned:

```
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
| The current date    | The date of today   | The Sunday of this week | The first date of this month | The first date of this quarter | The first date of this year |
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
| 2020-03-08 22:41:46 | 2020-03-08 00:00:00 | 2020-03-08 00:00:00 | 2020-03-01 00:00:00 | 2020-01-01 00:00:00 | 2020-01-01 00:00:00 |
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
```

In the following example, `ROUND` calculates the nearest dates that meet the different specified requirements when the input date is the same:

```
SELECT SYSDATE The current date,
ROUND(SYSDATE) The nearest date to 00:00,
ROUND(SYSDATE,'DAY') The nearest Sunday,
ROUND(SYSDATE,'MONTH') The first date of the nearest month,
ROUND(SYSDATE,'Q') The first date of the nearest quarter,
ROUND(SYSDATE,'YEAR') The first date of the nearest year FROM DUAL;
```

The following query result is returned:

```
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
| The current date            | The nearest date to 00:00          | The nearest Sunday          | The first date of the nearest month            | The first date of the nearest quarter         | The first date of the nearest year        |
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
| 2020-03-08 22:41:02 | 2020-03-09 00:00:00 | 2020-03-08 00:00:00 | 2020-03-01 00:00:00 | 2020-04-01 00:00:00 | 2020-01-01 00:00:00 |
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
```

TZ_OFFSET

The `TZ_OFFSET` function returns the time zone offset of the `n` time zone. The time zone offset is the difference (in hours and minutes) between a specified time zone and the UTC+0 time zone.

Syntax

```
TZ_OFFSET (n)
```

Parameters

Parameter	Description
n	The name of the time zone region.

Return type

`VARCHAR2` data type

Examples

The following example returns the time zone offsets of the current session time zone specified by `SESSIONTIMEZONE`, the database time zone specified by `DBTIMEZONE`, and the US/Eastern time zone:

```
SELECT TZ_OFFSET(SESSIONTIMEZONE),TZ_OFFSET(DBTIMEZONE),TZ_OFFSET('US/Eastern') FROM DUAL;
```

The following query result is returned:

```
+-----------------------------+-----------------------+-------------------------+
| TZ_OFFSET(SESSIONTIMEZONE) | TZ_OFFSET(DBTIMEZONE) | TZ_OFFSET('US/EASTERN') |
+-----------------------------+-----------------------+-------------------------+
| +08:00                      | +00:00                | -04:00                  |
+-----------------------------+-----------------------+-------------------------+
```

17.1.5.6.2.5. General comparison functions

GREATEST

The `GREATEST` function returns the maximum value in a list of one or more expressions. The database uses the first parameter `expr` to determine the return type. If the data type of the remaining parameters is different from that of the first parameter `expr`, ApsaraDB for OceanBase implicitly converts each parameter that follows the first parameter `expr` to the data type of the first parameter `expr` before the comparison.

Syntax

```
GREATEST(expr [, expr ]...)
```

Parameters

Parameter	Description
expr	An expression or an expression list. The data type can be `NUMBER`, `FLOAT`, `BINARY_FLOAT`, `BINARY_DOUBLE`, `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.

Return type

If the data type of the first parameter `expr` is `NUMBER`, `FLOAT`, `BINARY_FLOAT`, or `BINARY_DOUBLE`, the function returns the data type that is the same as the data type of the first parameter `expr`. If the data type of the first parameter `expr` is `CHAR`, `VARCHAR2`, or `CLOB`, the function returns the `VARCHAR2` type. If the data type of the first parameter `expr` is `NCHAR` or `NVARCHAR2`, the function returns the `NVARCHAR2` type.

Examples

The following statement compares the sizes of strings and returns the maximum string:

```
SELECT GREATEST('HAPPY', 'HAPPEN', 'HAPPINESS') "Greatest"
  FROM DUAL;
```

The following result is returned:

```
+-----------+
| Greatest |
+-----------+
|    HAPPY |
+-----------+
```

The following statement compares the sizes of the **1** integer, the **3.925** string, and the **2.4** string. The sizes can be compared after the remaining parameters are implicitly converted to the numeric data type. This is because the data type of the first parameter is the numeric data type.

```
SELECT GREATEST (1, '3.935', '2.4') "Greatest"
  FROM DUAL;
```

The following result is returned:

```
+-----------+
| Greatest |
+-----------+
|    3.935 |
+-----------+
```

LEAST

The `LEAST` function returns the minimum value in a list of one or more expressions. The database uses the first parameter `expr` to determine the return type. If the data type of the remaining parameters is different from that of the first parameter `expr`, ApsaraDB for OceanBase implicitly converts each parameter that follows the first parameter `expr` to the data type of the first parameter `expr` before the comparison.

Syntax

```
LEAST(expr [, expr ]...)
```

Parameters

Parameter	Description
expr	An expression or an expression list. The data type can be `NUMBER`, `FLOAT`, `BINARY_FLOAT`, `BINARY_DOUBLE`, `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.

Return type

If the data type of the first parameter `expr` is `NUMBER`, `FLOAT`, `BINARY_FLOAT`, or `BINARY_DOUBLE`, the function returns the data type that is the same as the data type of the first parameter `expr`. If the data type of the first parameter `expr` is `CHAR`, `VARCHAR2`, or `CLOB`, the function returns the `VARCHAR2` type. If the data type of the first parameter `expr` is `NCHAR` or `NVARCHAR2`, the function returns the `NVARCHAR2` type.

Examples

The following statement compares the sizes of strings and returns the minimum string:

```
SELECT LEAST('HAPPY', 'HAPPEN', 'HAPPINESS') "Least"
  FROM DUAL;
```

The following result is returned:

```
+--------+
| Least  |
+--------+
| HAPPEN |
+--------+
```

The following statement compares the sizes of the **1** integer, the **3.925** string, and the **2.4** string. The sizes can be compared after the remaining parameters are implicitly converted to the numeric data type. This is because the data type of the first parameter is the numeric data type.

```
SELECT LEAST (1, '3.925', '2.4') "Least"
  FROM DUAL;
```

The following result is returned:

```
+-------+
| Least |
+-------+
|     1 |
+-------+
```

17.1.5.6.2.6. Conversion functions

CAST

The `CAST` function explicitly converts the expression of a source data type to another data type.

Syntax

```
CAST (expr AS type_name )
```

Parameters

Parameter	Description
expr	The column name or the expression.
AS	Separates two parameters. The parameter before `AS` specifies the data to be processed. The parameter after `AS` specifies the data type to be converted to.
type_name	The data type is the built-in data type of ApsaraDB for OceanBase. For more information, see Built-in data types.

Return type

The return type is the same as the data type of `type_name` .

The following table lists which data types can be converted to other built-in data types.

	from BINARY_FLOAT, BINARY_DOUBLE	from CHAR, VARCHAR2	from NUMBER	[1] from DATETIME/INTERVAL	from RAW	from NCHAR, NVARCHAR2
to BINARY_FLOAT, BINARY_DOUBLE	yes	yes	yes	no	no	yes
to CHAR, VARCHAR2	yes	yes	yes	yes	yes	no
to NUMBER	yes	yes	yes	no	no	yes
to DATETIME, INTERVAL	no	yes	no	yes	no	no
to RAW	yes	yes	yes	no	yes	no
to NCHAR, NVARCHAR2	yes	no	yes	yes	yes	yes

[1] `To DATETIME/INTERVAL` data types include `DATE` , `TIMESTAMP` , `TIMESTAMP WITH TIMEZONE` , `INTERVAL DAY TO SECOND` , and `INTERVAL YEAR TO MONTH` .

Examples

Execute the following statement:

```
SELECT CAST('123' AS INT),CAST(1 AS VARCHAR2(10)),CAST('22-OCT-1997' AS TIMESTAMP WITH LOCAL TIME ZONE)
AS RESULT FROM DUAL;
```

The following query result is returned:

```
+------------------+----------------------+----------------------------+
| CAST('123'ASINT) | CAST(1ASVARCHAR2(10)) | RESULT                     |
+------------------+----------------------+----------------------------+
|              123 | 1                    | 1997-10-22 00:00:00.000000 |
+------------------+----------------------+----------------------------+
```

HEXTORAW

The `HEXTORAW` function converts a character that contains hexadecimal digits in the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type to the `RAW` data type.

Syntax

```
HEXTORAW (char)
```

Parameters

Parameter	Description
char	The hexadecimal string. The string type can be `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2`.

Return type

The data of the `RAW` data type is returned.

Examples

Execute the following statement:

```
select HEXTORAW('A123')  from dual;
```

The following query result is returned:

```
+------------------+
| HEXTORAW('A123') |
+------------------+
| A123             |
+------------------+
```

RAWTOHEX

The `RAWTOHEX` function converts a binary number to a corresponding hexadecimal string.

Syntax

```
RAWTOHEX (raw)
```

Parameters

Parameter	Description
raw	The binary string.

Return type

The hexadecimal string

Examples

Execute the following statement:

```
SELECT RAWTOHEX('AB')  FROM DUAL;
```

The following query result is returned:

```
+-----------------+
| RAWTOHEX('AB') |
+-----------------+
| 4142           |
+-----------------+
```

TO_BINARY_DOUBLE

The `TO_BINARY_DOUBLE` function returns a 64-bit double-precision floating-point number.

Syntax

```
TO_BINARY_DOUBLE(expr [, fmt [, 'nlsparam' ] ])
```

Parameters

Parameter	Description
expr	The string or the numeric data type: `NUMBER`, `BINARY_FLOAT`, or `BINARY_DOUBLE`.

The optional parameters `fmt` and `nlsparam` are valid only if `expr` is a string. They serve the same purpose as the `TO_CHAR(number)` feature. If `expr` is `BINARY_DOUBLE`, the function returns `expr`.

Return type

A 64-bit double-precision floating-point number is returned. The result of converting a string or `NUMBER` to `BINARY_DOUBLE` may be inexact. This is because `NUMBER` and character types use decimal precision to represent numeric values, but `BINARY_DOUBLE` uses binary precision. The result of converting `BINARY_FLOAT` to `BINARY_DOUBLE` is exact.

Examples

Execute the following statement:

```
SELECT TO_BINARY_DOUBLE(1222.111) FROM DUAL;
```

The following query result is returned:

```
+----------------------------+
| TO_BINARY_DOUBLE(1222.111) |
+----------------------------+
|              1.222111E+003 |
+----------------------------+
```

You can refer to TO_BINARY_FLOAT and TO_CHAR(number).

TO_BINARY_FLOAT

The `TO_BINARY_FLOAT` function returns a 32-bit single-precision floating-point number.

Syntax

```
TO_BINARY_FLOAT(expr [, fmt [, 'nlsparam' ] ])
```

Parameters

Parameter	Description
expr	The string or the numeric data type: `NUMBER`, `BINARY_FLOAT`, or `BINARY_DOUBLE`.

The optional parameters `fmt` and `nlsparam` are valid only if `expr` is a string. They serve the same purpose as the `TO_CHAR(number)` feature. If `expr` is `BINARY_FLOAT`, the function returns `expr`.

Return type

A 32-bit single-precision floating-point number is returned. The result of converting a string or `NUMBER` to `BINARY_FLOAT` may be inexact. This is because `NUMBER` and character types use decimal precision to represent numeric values, but `BINARY_FLOAT` uses binary precision. If a `BINARY_DOUBLE` value uses higher precision than the precision that is supported by `BINARY_FLOAT`, the result of converting `BINARY_DOUBLE` to `BINARY_FLOAT` is inexact.

Examples

Execute the following statement:

```
SELECT TO_BINARY_FLOAT(1222.111) from dual;
```

The following query result is returned:

```
+---------------------------+
| TO_BINARY_FLOAT(1222.111) |
+---------------------------+
|           1.22211096E+003 |
+---------------------------+
```

You can refer to TO_BINARY_DOUBLE and TO_CHAR(number).

TO_CHAR (character)

The `TO_CHAR (character)` function converts the data of `NCHAR`, `NVARCHAR2`, or `CLOB` to a database character set.

Syntax

```
TO_CHAR(character )
```

Parameter

	Description
character	The data type can be `NCHAR`, `NVARCHAR2`, or `CLOB`.

Return type

The return type is `VARCHAR2`. When the function converts a character `CLOB` to a database character set, the database returns an error if the `CLOB` value to be converted is greater than the desired type.

Examples

Create the **CLOBTEST** table and insert data to the column of the `RAW` data type.

```
CREATE TABLE CLOBTEST(TEXT CLOB);
INSERT INTO CLOBTEST VALUES('DWUIDBWUIDBWIOBFWUIOBFIOWBFWUIOBFUWIFB') ;
```

Execute the following statements:

```
SELECT TO_CHAR(TEXT) FROM CLOBTEST;
```

The following query result is returned:

```
+----------------------------------------+
| TO_CHAR(TEXT)                          |
+----------------------------------------+
| dwuidbwuidbwiobfwuiobfiowbfwuiobfuwifb |
+----------------------------------------+
```

TO_CHAR (datetime)

The `TO_CHAR` function converts a value of a data type to a value of the `VARCHAR2` data type based on the format that is specified by the `fmt` parameter. The data types of which values can be converted include `DATE`, `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE`, `TIMESTAMP WITH LOCAL TIME ZONE`, `INTERVAL DAY TO SECOND`, and `INTERVAL YEAR TO MONTH`. If you do not specify the `fmt` parameter, the value of the `datetime` parameter is converted to a value of the `VARCHAR2` data type based on the following formats.

- The values of `DATE`, `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE`, and `TIMESTAMP WITH LOCAL TIME ZONE` are converted to the values that are in the default format of date and time values in the database. In the Data types topic, you can view the default format for each date and time type.
- The values of the `INTERVAL DAY TO SECOND` and `INTERVAL YEAR TO MONTH` data types are converted to the interval values in the numeric format.

Syntax

```
TO_CHAR({ datetime | interval } [, fmt [, 'nlsparam' ] ])
```

Parameters

Parameter	Description
datetime	The values of the data types, such as `DATE`, `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE`, `TIMESTAMP WITH LOCAL TIME ZONE`, `INTERVAL DAY TO SECOND`, and `INTERVAL YEAR TO MONTH`.
fmt	The output format parameter.
nlsparam	Controls the language of the returned month and day.

Return type

`VARCHAR2` data type

Examples

Example 1: The following statement returns the current system date and converts the date and time value to the value in the **DS DL** format by using the TO_CHAR function.

```
SELECT TO_CHAR(SYSDATE,'DS DL') FROM DUAL;
```

The following query result is returned:

```
+-----------------------------------+
| TO_CHAR(SYSDATE,'DSDL')           |
+-----------------------------------+
| 03/08/2020 Sunday, March 08, 2020 |
+-----------------------------------+
```

Example 2: In the following statement, the interval value is converted to the value in the specified format and the language of the returned result is specified as **AMERICAN**.

```
SELECT TO_CHAR(interval'1' year, 'SS-MI-HH', 'nls_language = AMERICAN') FROM DUAL;
```

The following query result is returned:

```
+-------------------------------------------------------------+
| TO_CHAR(INTERVAL'1'YEAR,'SS-MI-HH','NLS_LANGUAGE=AMERICAN') |
+-------------------------------------------------------------+
| +01-00                                                      |
+-------------------------------------------------------------+
```

TO_CHAR (number)

The `TO_CHAR` function converts the numeric value `n` of the `NUMBER`, `BINARY_FLOAT`, or `BINARY_DOUBLE` type to a value of the `varchar2` data type based on the specified numeric format `fmt`. If you omit `fmt`, `n` is converted to a `VARCHAR2` value and the value is long enough to hold valid digits. If `n` is negative, the negative sign is displayed on the leftmost side of the output value. For example, `TO_CHAR(-1,'$9')` returns `-$1` instead of `$-1`.

Syntax

```
TO_CHAR(n [, fmt [, 'nlsparam' ] ])
```

Parameters

Parameter	Description
n	The expression of exact numeric data types or approximate numeric data types. The numeric data type can be `NUMBER`, `BINARY_FLOAT`, or `BINARY_DOUBLE`.

Parameter	Description
fmt	The output format parameter.
nlsparam	The supported language is obtained from `sys.V_$NLS_VALID_VALUES`.

fmt parameter list

fmt parameter	Description
9	Returns the value for which the number of digits is specified.
0	It returns leading zeros. It returns trailing zeros.
, (Comma)	Returns the comma in the specified position. You can specify multiple commas when you format the number. **Limits**: You cannot start to format a numeric value from a comma, and the comma cannot appear to the right of a decimal character or a period.
. (Decimal point)	Returns a decimal where the decimal point is in the specified position. **Limits**: You can specify only one decimal point when you format the number.

Return type

`VARCHAR2` data type

Examples

Execute the following statement:

```
SELECT TO_CHAR(123.456,'999') FROM DUAL;
```

The following query result is returned:

```
+------------------------+
| TO_CHAR(123.456,'999') |
+------------------------+
|  123                   |
+----------------------
```

TO_DATE

The `TO_DATE` function converts a character of the `CHAR`, `VARCHAR`, `NCHAR`, or `NVARCHAR2` data type to a value of the date data type.

Syntax

```
TO_DATE(char [, fmt [, 'nlsparam' ] ])
```

Parameters

Parameter	Description
char	The data of the `CHAR`, `VARCHAR`, `NCHAR`, or `NVARCHAR2` character type.
fmt	The date and time format.
nlsparam	The value is implicitly specified by the `nls_territory` initialization parameter or explicitly specified by the `nls_date_format` parameter.

Return type

The return type is the `DATE` type.

Examples

Execute the following statement:

```
SELECT TO_DATE('199912','YYYYMM'),TO_DATE('2000.05.20','YYYY.MM.DD'),
(DATE '2008-12-31') XXDATE,
TO_DATE('2008-12-31 12:31:30','YYYY-MM-DD HH24:MI:SS'),
(TIMESTAMP '2008-12-31 12:31:30') XXTIMESTAMP
FROM DUAL;
```

The following query result is returned:

```
+---------------------------+-----------------------------------+---------------------+------------
-----------------------------------------+-------------------------------+
| TO_DATE('199912','YYYYMM') | TO_DATE('2000.05.20','YYYY.MM.DD') | XXDATE              | TO_DATE('20
08-12-3112:31:30','YYYY-MM-DDHH24:MI:SS') | XXTIMESTAMP                   |
+---------------------------+-----------------------------------+---------------------+------------
-----------------------------------------+-------------------------------+
| 1999-12-01 00:00:00        | 2000-05-20 00:00:00               | 2008-12-31 00:00:00 | 2008-12-31
12:31:30                                | 2008-12-31 12:31:30.000000000 |
+---------------------------+-----------------------------------+---------------------+------------
-----------------------------------------+-------------------------------+
```

TO_DSINTERVAL

The `TO_DSINTERVAL` function converts a string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type to a value of the `INTERVAL DAY TO SECOND` data type. You can use this function to add or subtract a date and time value.

Syntax

```
TO_DSINTERVAL (days hours:minutes:seconds[.frac_secs])
```

Parameters

Parameter	Description
days hours:minutes:seconds[.frac_secs]	The string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type in compliance with the format of this parameter.

Return type

`INTERVAL DAY TO SECOND` data type

Examples

The following example returns a date and time value 100 days later than the current time:

```
SELECT SYSDATE, SYSDATE+TO_DSINTERVAL('100 00:00:00') FROM DUAL;
```

The following query result is returned:

```
+---------------------+---------------------------------------+
| SYSDATE             | SYSDATE+TO_DSINTERVAL('10000:00:00') |
+---------------------+---------------------------------------+
| 2020-03-26 12:40:39 | 2020-07-04 12:40:39                   |
+---------------------+---------------------------------------+
```

TO_NUMBER

The `TO_NUMBER` function converts `expr` to a value of the numeric data type. `expr` can be a value of the `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, `BINARY_FLOAT`, or `BINARY_DOUBLE` data type.

Syntax

```
TO_NUMBER(expr [, fmt [, 'nlsparam' ] ])
```

Parameters

Parameter	Description

Parameter	Description
expr	The value of the `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, `BINARY_FLOAT`, or `BINARY_DOUBLE` data type.
fmt	The format model.
nlsparam	The supported language is obtained from `sys.V_$NLS_VALID_VALUES`.

Return type

The data of the `NUMBER` type is returned.

Examples

Execute the following statement:

```
SELECT TO_NUMBER('199912'),TO_NUMBER('450.05') FROM DUAL;
```

The following query result is returned:

```
+---------------------+---------------------+
| TO_NUMBER('199912') | TO_NUMBER('450.05') |
+---------------------+---------------------+
|              199912 |              450.05 |
+---------------------+---------------------+
```

TO_TIMESTAMP

The `TO_TIMESTAMP` function converts a string to the `TIMESTAMP` data type.

Syntax

```
TO_TIMESTAMP (char,[fmt],['nlsparam'])
```

Parameters

Parameter	Description
char	The string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type.

Parameter	Description
fmt	Specifies the format of the return value.
nlsparam	Controls the language of the returned month and day.

Return type

`TIMESTAMP` data type

Examples

Execute the following statement:

```
SELECT TO_TIMESTAMP ('10-Sep-02 14:10:10.123000', 'DD-Mon-RR HH24:MI:SS.FF')
FROM DUAL;
```

The following query result is returned:

```
+---------------------------------------------------------------------+
| TO_TIMESTAMP('10-SEP-0214:10:10.123000','DD-MON-RRHH24:MI:SS.FF') |
+---------------------------------------------------------------------+
| 2002-09-10 14:10:10.123000000                                       |
+---------------------------------------------------------------------+
```

TO_TIMESTAMP_TZ

The `TO_TIMESTAMP` function converts a string to the `TIMESTAMP WITH TIME ZONE` data type, including information about time zones.

Syntax

```
TO_TIMESTAMP_TZ (char,[fmt],['nlsparam'])
```

Parameters

Parameter	Description
char	The string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type.
fmt	Specifies the output format.
nlsparam	Controls the language of the returned month and day.

Return type

`TIMESTAMP WITH TIME ZONE` data type

Examples

Execute the following statement:

```
SELECT TO_TIMESTAMP_TZ ('10-Sep-02 14:10:10.123000', 'DD-Mon-RR HH24:MI:SS.FF') FROM DUAL;
```

The following query result is returned:

```
+---------------------------------------------------------------------+
| TO_TIMESTAMP_TZ('10-SEP-0214:10:10.123000','DD-MON-RRHH24:MI:SS.FF') |
+---------------------------------------------------------------------+
| 2002-09-10 14:10:10.123000000 +08:00                                |
+---------------------------------------------------------------------+
```

TO_YMINTERVAL

The `TO_YMINTERVAL` function converts a string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type to a value of the `INTERVAL YEAR TO MONTH` data type. You can use this function to add or subtract a date and time value.

Syntax

```
TO_YMINTERVAL (years-months)
```

Parameters

Parameter	Description
years-months	The string of the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` data type in compliance with the format of this parameter.

Return type

`INTERVAL YEAR TO MONTH` data type

Examples

The following example shows that the date and time value one year and two months later than the current time is returned:

```
SELECT SYSDATE,SYSDATE+TO_YMINTERVAL('01-02') FROM DUAL;
```

The following query result is returned:

```
+---------------------+--------------------------------+
| SYSDATE             | SYSDATE+TO_YMINTERVAL('01-02') |
+---------------------+--------------------------------+
| 2020-03-08 22:32:01 | 2021-05-08 22:32:01            |
+---------------------+--------------------------------+
```

17.1.5.6.2.7. Encoding and decoding functions

DECODE

The `DECODE` function compares the `search` parameter with `condition` in sequence until the value of `condition` is equal to that of `search`. Then, the function returns the value of the `result` parameter that follows the corresponding `search`. If no `search` is equal to `condition`, the function returns the value of the `default` parameter.

Syntax

```
DECODE (condition, search 1, result 1, search 2, result 2 ... search n, result n, default)
```

The meaning of the `DECODE` function can be interpreted by using the `IF...ELSE IF...END` statement.

```
IF condition = search 1 THEN
RETURN(result 1)
ELSE IF condition = search 2 THEN
RETURN(result 2)
......
ELSE IF condition = search n THEN
RETURN(result n)
ELSE
RETURN(default)
END IF
```

Parameters

Parameter	Description
condition, search 1...search n, result 1...result n, default	The expression of the `NUMBER`, `FLOAT`, `BINARY_FLOAT`, or `BINARY_DOUBLE` numeric type or the `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` character type.

> **Notice**
>
> `search 1` to `search n` cannot be conditional expressions. In this case, you can execute only the `CASE WHEN THEN END` statement to resolve the issue.

```
WHEN CASE condition = search 1 THEN
RETURN(result 1)
ELSE CASE condition = search 2 THEN
RETURN(result 2)
......
ELSE CASE condition = search n THEN
RETURN(result n)
ELSE
RETURN(default)
END
```

Examples

Example 1: Use `DECODE` to compare numeric values.

The following statement uses the `DECODE` function to return a smaller number between 10 and 20. The `SIGN()` function is used to calculate the sign of the difference between two values. The difference is a negative number and `SIGN()` returns -1 because 10 is less than 20. In this case, the `DECODE` function compares the -1 argument with the return value of the `SIGN()` function. If they are equal, 10 is returned. If they are not equal, 20 is returned.

```
SELECT DECODE(SIGN(10-20),-1,10,20) FROM DUAL;
```

The following result is returned:

```
+------------------------------+
| DECODE(SIGN(10-20),-1,10,20) |
+------------------------------+
|                           10 |
+------------------------------+
```

Example 2: Run the `DECODE` function to check whether the data includes the **S** character.

The following statements create the **EMP** table that contains the **ename** and **sal** columns and insert values into the table:

```
CREATE TABLE EMP(ename VARCHAR(30),sal NUMBER);
INSERT INTO EMP VALUES('CLARK', 2750);
INSERT INTO EMP VALUES('KING', 5300);
INSERT INTO EMP VALUES('MILLER', 1600);
INSERT INTO EMP VALUES('ADAMS', 1400);
INSERT INTO EMP VALUES('FORD', 3300);
INSERT INTO EMP VALUES('JONES', 3275);
INSERT INTO EMP VALUES('SCOTT', 3300);
INSERT INTO EMP VALUES('SMITH', 1100);
INSERT INTO EMP VALUES('ALLEN', 1900);
INSERT INTO EMP VALUES('BLAKE', 3150);
INSERT INTO EMP VALUES('JAMES', 1250);
INSERT INTO EMP VALUES('MARTIN', 1550);
INSERT INTO EMP VALUES('TURNER', 1800);
INSERT INTO EMP VALUES('WARD', 1550);
```

The following statement returns the occurrence position of the **S** character in the values of the **ename** column by using the `INSTR()` function. If the S character does not occur, 0 is returned. In this case, the `DECODE` function compares the return value of the `INSTR` function with 0. If they are equal, the **S** character does not occur in the values and the `DECODE` function returns **S excluded**. Otherwise, it returns **S included**.

```
SELECT ENAME, SAL, DECODE(INSTR(ename, 'S'), 0, 'S excluded', 'S included') AS INFO FROM EMP;
```

The following query result is returned:

```
+--------+------+------------+
| ENAME  | SAL  | INFO       |
+--------+------+------------+
| CLARK  | 2750 | S excluded    |
| KING   | 5300 | S excluded    |
| MILLER | 1600 | S excluded    |
| ADAMS  | 1400 | S included      |
| FORD   | 3300 | S excluded    |
| JONES  | 3275 | S included      |
| SCOTT  | 3300 | S included      |
| SMITH  | 1100 | S included      |
| ALLEN  | 1900 | S excluded    |
| BLAKE  | 3150 | S excluded    |
| JAMES  | 1250 | S included      |
| MARTIN | 1550 | S excluded    |
| TURNER | 1800 | S excluded    |
| WARD   | 1550 | S excluded    |
+--------+------+------------+
```

ORA_HASH

The `ORA_HASH` function retrieves the hash value of a corresponding expression.

Syntax

```
ORA_HASH(expr [, max_bucket [, seed_value ] ])
```

Parameters

Parameter	Description
expr	The value is generally a column name in the database table. The data type can be the numeric type, character type, date and time type, or `RAW` type.
max_bucket	The optional `max_bucket` parameter determines the maximum number of buckets that the hash function can return. Valid values: 0 to 4294967295. Default value: 4294967295.
seed_value	The optional `seed_value` parameter enables ApsaraDB for OceanBase to produce a number of different results for the same group of data. You can specify a value from 0 to 4294967295. Default value: 0.

Return type

Data of the `NUMBER` type

Examples

To create the **SALE** table and insert data into the table, execute the following statements:

```
CREATE TABLE SALE(MONTH CHAR(6), SELL NUMBER(10,2));
INSERT INTO SALE VALUES(200001, 1000);
INSERT INTO SALE VALUES(200002, 1100);
INSERT INTO SALE VALUES(200003, 1200);
INSERT INTO SALE VALUES(200004, 1300);
INSERT INTO SALE VALUES(200005, 1400);
INSERT INTO SALE VALUES(200006, 1500);
INSERT INTO SALE VALUES(200007, 1600);
INSERT INTO SALE VALUES(200101, 1100);
INSERT INTO SALE VALUES(200202, 1200);
INSERT INTO SALE VALUES(200301, 1300);
```

To use the `ORA_HASH` function to query the **SALE** table, execute the following statement:

```
SELECT ORA_HASH(CONCAT(month,sell),12,0), month, sell FROM Sale;
```

The following query result is returned:

```
+-----------------------------------+--------+------+
| ORA_HASH(CONCAT(MONTH,SELL),12,0) | MONTH  | SELL |
+-----------------------------------+--------+------+
|                                 1 | 200001 | 1000 |
|                                 6 | 200002 | 1100 |
|                                 5 | 200003 | 1200 |
|                                 4 | 200004 | 1300 |
|                                 5 | 200005 | 1400 |
|                                 2 | 200006 | 1500 |
|                                 7 | 200007 | 1600 |
|                                10 | 200101 | 1100 |
|                                 7 | 200202 | 1200 |
|                                 4 | 200301 | 1300 |
+-----------------------------------+--------+------+
```

VSIZE

The `VSIZE` function returns the number of bytes for `X`.

Syntax

```
VSIZE(X)
```

Return type

The number of bytes for x is returned. If x is `NULL`, the function returns `NULL`.

Examples

To create the **employees** table and insert data to the table, execute the following statements:

```
CREATE TABLE employees(manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(300, 'Wei',  '2019-09-11',23600);
INSERT INTO employees VALUES(200, 'Red', '2019-11-05', 23800);
INSERT INTO employees VALUES(100, 'Part',  '2018-10-01',24000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',23500);
COMMIT;
```

To use the `VSIZE` function to query the number of bytes for **manager_id = 300** in the **last_name** column, execute the following statement:

```
SELECT last_name, VSIZE (last_name) "BYTES" FROM employees WHERE manager_id = 300;
```

The following query result is returned:

```
+-----------+-------+
| LAST_NAME | BYTES |
+-----------+-------+
| Wei       |     3 |
+-----------+-------+
```

17.1.5.6.2.8. Null value-related functions

COALESCE

The `COALESCE` function returns the first non-null expression in a parameter list. You must specify at least two parameters.

Syntax

```
COALESCE(expr1, expr2[,…, exprn])
```

Parameters

Parameter	Description
expr1, expr2[,..., exprn]	The non-null expressions. You must specify at least two non-null expressions.

Return type

The first non-null expression in the parameter list is returned. If all the parameters are `NULL`, `NULL` is returned.

Examples

Assume that a **product_information** table is available. In the table, **product_id** indicates a product ID, **list_price** indicates the original price of the product, **min_price** indicates the lowest price of the product, and **Sale** indicates the actual sale price of the product. Specify the product discount as 10% and calculate the actual sale price of each product. In this case, you can use the `COALESCE` function. If **list_price** is empty, perform calculations based on **min_price**. If **min_price** is also empty, perform calculations based on **5**.

You can execute the following statements to create the **product_information** data table and insert data into the table:

```
CREATE TABLE product_information(supplier_id INT, product_id INT,list_price numeric, min_price numeri
c);
INSERT INTO PRODUCT_INFORMATION VALUES ('102050', '1659', '45', NULL);
INSERT INTO PRODUCT_INFORMATION VALUES ('102050', '1770', NULL, '70');
INSERT INTO PRODUCT_INFORMATION VALUES ('102050', '2370', '305', '247');
INSERT INTO PRODUCT_INFORMATION VALUES ('102050', '2380', '750', '731');
INSERT INTO PRODUCT_INFORMATION VALUES ('102050', '3255', NULL, NULL);
```

Execute the following query statement:

```
SELECT product_id, list_price,min_price,COALESCE(0.9*list_price, min_price, 5) "Sale"
FROM product_information WHERE supplier_id = 102050 ORDER BY product_id;
```

The following query result is returned:

```
+--------------+--------------+------------+--------+
|  PRODUCT_ID  |  LIST_PRICE  |  MIN_PRICE |  Sale  |
+--------------+--------------+------------+--------+
|        1659  |       45     |            |  40.5  |
+--------------+--------------+------------+--------+
|        1770  |              |     70     |    70  |
+--------------+--------------+------------+--------+
|        2370  |      305     |    247     | 274.5  |
+--------------+--------------+------------+--------+
|        2380  |      750     |    731     |   675  |
+--------------+--------------+------------+--------+
|        3255  |              |            |     5  |
+--------------+--------------+------------+--------+
```

LNNVL

The `LNNVL` function determines whether one or two operands in a condition are `NULL`. You can use this function in a `WHERE` clause or use the function as the `WHEN` condition in a `CASE` expression. A condition is used as a parameter. If the condition is `FALSE` or `UNKNOWN`, `TRUE` is returned. If the condition is `TRUE`, `FALSE` is returned.

Syntax

```
LNNVL(condition)
```

Parameters

Parameter	Description
condition	The condition.

Assume that a is equal to 2 and the value of b is **NULL**. The following table lists the return values of the `LNNVL` function.

Condition	Check result of the condition	Return value of LNNVL
a = 1	FALSE	TRUE
a = 2	TRUE	FALSE
a IS NULL	FALSE	TRUE
b = 1	UNKNOWN	TRUE
b IS NULL	TRUE	FALSE

Condition	Check result of the condition	Return value of LNNVL
a = b	UNKNOWN	TRUE

Return type

`TRUE` or `FALSE` of the BOOLEAN type is returned.

Examples

Assume that data is inserted into the employee name column **name** and the commission column **commission_pct** in an **EMPLOYEES** table. Execute the following statements:

```
CREATE TABLE EMPLOYEES (name VARCHAR(20), commission_pct numeric);
INSERT INTO EMPLOYEEs VALUES ('Baer', null);
INSERT INTO EMPLOYEEs VALUES ('Bada', null);
INSERT INTO EMPLOYEEs VALUES ('Boll', 0.1);
INSERT INTO EMPLOYEEs VALUES ('Bates', 0.15);
INSERT INTO EMPLOYEEs VALUES ('Eros', null);
INSERT INTO EMPLOYEEs VALUES ('Girl', 0.25);
```

You want to know the number of employees whose commission rates are less than 20%, including the employees who do not receive commissions. You can query only the number of employees whose actual commission rates are less than 20% by executing the following statement:

```
SELECT COUNT(*) FROM employees WHERE commission_pct < .2;
```

The following query result is returned:

```
+------------+
|  COUNT(*)  |
+------------+
|        2   |
+------------+
```

To include another three employees who do not receive commissions, you must rewrite the query by using the `INNVL` function. Execute the following statement:

```
SELECT COUNT(*) FROM employees WHERE LNNVL(commission_pct >= .2);
```

The following query result is returned:

```
+------------+
|  COUNT(*)  |
+------------+
|      4     |
+------------+
```

NVL

The `NVL` function returns a non-`NULL` value from two expressions. If the results of `expr1` and `expr2` are `NULL` values, the `NVL` function returns `NULL`.

Syntax

```
NVL(expr1, expr2)
```

Parameters

Parameter	Description
expr1	The expression. The data type can be one of the built-in data types of ApsaraDB for OceanBase.
expr2	The expression. The data type can be one of the built-in data types of ApsaraDB for OceanBase.

`expr1` and `expr2` must be the same type or can be implicitly converted to the same type. If they cannot be implicitly converted, ApsaraDB for OceanBase returns an error. Implicit conversions are implemented in the following ways:

- If `expr1` is the data of the `CHAR`, `NCHAR`, `NVARCHAR`, `VARCHAR2`, or `VARCHAR` character type, ApsaraDB for OceanBase converts `expr2` to the data type of `expr1` before it compares `expr1`. Then, ApsaraDB for OceanBase returns `VARCHAR2` in the character set of `expr1`.
- If `expr1` is the data of the `NUMBER`, `FLOAT`, `BINARY_FLOAT`, or `BINARY_DOUBLE` numeric type, ApsaraDB for OceanBase determines the parameter that has the highest numeric precedence. ApsaraDB for OceanBase implicitly converts the other parameter to this data type and returns the data type.

Return type

If `expr1` and `expr2` are `NULL`, `NULL` is returned. If `expr1` is the data of the `CHAR`, `NCHAR`, `NVARCHAR`, `VARCHAR2`, or `VARCHAR` character type, `VARCHAR2` in the character set of `expr1` is returned. If `expr1` is the data of the `NUMBER`, `FLOAT`, `BINARY_FLOAT`, or `BINARY_DOUBLE` numeric type, the data type that has the highest numeric precedence in `expr1` is returned.

Examples

Assume that data is inserted into the employee name column **name** and the commission column **commission_pct** in an **EMPLOYEES** table. Execute the following statements:

```
CREATE TABLE EMPLOYEES (name VARCHAR(20),commission_pct float(3));
INSERT INTO EMPLOYEEs VALUES ('Baer', null);
INSERT INTO EMPLOYEEs VALUES ('Bada', null);
INSERT INTO EMPLOYEEs VALUES ('Boll', 0.1);
INSERT INTO EMPLOYEEs VALUES ('Bates', 0.15);
INSERT INTO EMPLOYEEs VALUES ('Eric', null);
```

Query the name and the commission of an employee. If the employee does not receive the commission, **Not Applicable** appears. Execute the following statement:

```
SELECT name, NVL(TO_CHAR(commission_pct), 'Not Applicable') commission
FROM employees WHERE name LIKE 'B%' ORDER BY name;
```

The following query result is returned:

```
+--------------+----------------+
|        NAME  |     COMMISSION |
+--------------+----------------+
|        Baer  | Not Applicable |
+--------------+----------------+
|        Bada  | Not Applicable |
+--------------+----------------+
|        Boll  |             .1 |
+--------------+----------------+
|       Bates  |            .15 |
+--------------+----------------+
```

NVL2

The `NVL2` function returns different values based on whether an expression is null. If `expr1` is not null, the value of `expr2` is returned. If `expr1` is null, the value of `expr3` is returned. If the types of `expr2` and `expr3` are different, `expr3` is converted to the type of `expr1`.

Syntax

```
NVL2(expr1, expr2, expr3)
```

Parameters

Parameter	Description
expr1	The expression. The data type can be one of the built-in data types of ApsaraDB for OceanBase.
expr2	The expression. The data type can be one of the built-in data types of ApsaraDB for OceanBase.
expr3	The expression. The data type can be one of the built-in data types of ApsaraDB for OceanBase.

If the data types of `expr2` and `expr3` are different, ApsaraDB for OceanBase implicitly converts one data type to the other data type. If the data types cannot be implicitly converted, the database returns an error. If `expr2` is character or numeric data, the following implicit conversion rules are used:

- If `expr2` is the data of the `CHAR`, `NCHAR`, `NVARCHAR`, `VARCHAR2`, or `VARCHAR` character type, ApsaraDB for OceanBase converts `expr3` to the data type of `expr2` before the value is returned unless `expr3` is `NULL`. In this case, the data types do not need to be converted, and the database returns `VARCHAR2` in the character set of `expr2`.
- If `expr2` is the data of the `NUMBER`, `FLOAT`, `BINARY_FLOAT`, or `BINARY_DOUBL` numeric type, ApsaraDB for OceanBase determines the parameter that has the highest numeric precedence. ApsaraDB for OceanBase implicitly converts the other parameter to this data type and returns the data type.

Return type

If `expr1` and `expr2` are `NULL`, `NULL` is returned. If `expr1` is the data of the `CHAR`, `NCHAR`, `NVARCHAR`, `VARCHAR2`, or `VARCHAR` character type, `VARCHAR2` in the character set of `expr1` is returned. If `expr1` is the data of the `NUMBER`, `FLOAT`, `BINARY_FLOAT`, or `BINARY_DOUBLE` numeric type, the data type that has the highest numeric precedence in `expr1` is returned.

Examples

Assume that data is inserted into the employee name column **name**, the salary name **salary**, and the commission column **commission_pct** in the **EMPLOYEES** table. Execute the following statements:

```
CREATE TABLE EMPLOYEES (name VARCHAR(20),commission_pct numeric);
INSERT INTO EMPLOYEEs VALUES ('Baer', 10000, null);
INSERT INTO EMPLOYEEs VALUES ('Bada', 2800, null);
INSERT INTO EMPLOYEEs VALUES ('Boll', 5600, .25);
INSERT INTO EMPLOYEEs VALUES ('Bates', 7300, .39);
INSERT INTO EMPLOYEEs VALUES ('Broll', 4000, null);
```

Use the `NVL2` function to query the total income of employees. If the **commission_pct** column is not empty for an employee, the income of the employee consists of the salary and the commission. Otherwise, the income of the employee is only the salary. Execute the following statement:

```
SELECT name, salary,NVL2(commission_pct, salary + (salary * commission_pct), salary) income
FROM employees WHERE name like 'B%' ORDER BY name;
```

The following query result is returned:

```
+--------------+----------------+----------------+
|        NAME  |        SALARY  |        INCOME  |
+--------------+----------------+----------------+
|        Bear  |         10000  |         10000  |
+--------------+----------------+----------------+
|        Bada  |          2800  |          2800  |
+--------------+----------------+----------------+
|        Boll  |          5600  |          7280  |
+--------------+----------------+----------------+
|       Bates  |          7300  |         10220  |
+--------------+----------------+----------------+
|       Broll  |          4000  |          4000  |
+--------------+----------------+----------------+
```

17.1.5.6.3. Aggregate functions

17.1.5.6.3.1. AVG

The `AVG` function returns the average value of a numeric column.

Syntax

```
AVG([ DISTINCT | UNIQUE | ALL ] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
DISTINCT	Removes duplicate values from the data and ignores NULL values in the data during the query.
UNIQUE	Removes duplicate values from the data and ignores NULL values in the data during the query.
ALL	Retains duplicate values in the data and ignores NULL values in the data during the query. Default value: `ALL` .
expr	The expression of the numeric type or the types that can be converted to the numeric type. The numeric type can be `NUMBER` , `FLOAT` , `BINARY_FLOAT` , or `BINARY_DOUBLE` .
OVER	Uses the `OVER` clause to define a window for calculation.

Notice

If you specify the `DISTINCT` or `UNIQUE` keyword, `order_by_clause` and `windowing_clause` cannot appear in `analytic_clause` .

Return type

The return type is the same as the data type of the `expr` parameter.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees (manager_id INT, last_name varchar(50), hiredate varchar(50), SALARY INT);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(100, 'Partners',  '2018-12-01',14000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',13500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 13000);
INSERT INTO employees VALUES(200, 'Part',  '2018-08-11',14000);
COMMIT;
```

Execute the following statement to calculate the average value of each column:

```
SELECT manager_id, last_name, hiredate, salary, AVG(salary) OVER (PARTITION BY manager_id
ORDER BY hiredate ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING) AS c_mavg
FROM employees ORDER BY manager_id, hiredate, salary;
```

The following result is returned:

```
+------------+-----------+------------+---------+-----------------------------------------+
| MANAGER_ID | LAST_NAME | HIREDATE   | SALARY | C_MAVG                                   |
+------------+-----------+------------+---------+-----------------------------------------+
|        100 | Errazuriz | 2017-07-21 |   1400 |                                   1550 |
|        100 | Raphaely  | 2017-07-22 |   1700 |                                   4700 |
|        100 | De Haan   | 2018-05-01 |  11000 |                                   8900 |
|        100 | Partners  | 2018-12-01 |  14000 |                                  13000 |
|        100 | Hartstein | 2019-05-01 |  14000 |                                13833.333 |
|        100 | Weiss     | 2019-07-11 |  13500 |                                  13500 |
|        100 | Russell   | 2019-10-05 |  13000 |                                  13250 |
|        200 | Part      | 2018-08-11 |  14000 |                                  13500 |
|        200 | Bell      | 2019-05-25 |  13000 |                                  13500 |
|        200 | Ross      | 2019-06-11 |  13500 |                                  13250 |
+------------+-----------+------------+---------+-----------------------------------------+
```

Examples of the aggregate function

Execute the following statement to calculate the average value of **salary**:

```
SELECT AVG(salary) FROM employees;
```

The following query result is returned:

```
+-----------------------------------------+
| AVG(SALARY)                             |
+-----------------------------------------+
| 10072.727272727272727272727272727273 |
+-----------------------------------------+
```

17.1.5.6.3.2. COUNT

The `COUNT` function queries the number of rows for `expr`.

Syntax

```
COUNT({ * | [ DISTINCT | UNIQUE | ALL ] expr }) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
*	All the rows that meet the conditions and include the rows whose values are NULL.
DISTINCT	Removes duplicate rows from returned rows and ignores the rows whose values are NULL.
UNIQUE	Removes duplicate rows from returned rows and ignores the rows whose values are NULL.
ALL	Returns all the values, including duplicate rows, and ignores the rows whose values are NULL.
expr	The expression of the numeric type or the types that can be converted to the numeric type. The numeric type can be `NUMBER`, `FLOAT`, `BINARY_FLOAT`, or `BINARY_DOUBLE`.
OVER	Uses the OVER clause to define a window for calculation.

> **Notice**
>
> - The `COUNT` function never returns `NULL`. If you specify `expr`, the function returns the number of rows where `expr` is not `NULL`. If you specify `COUNT(*)`, the function returns the number of all the rows. If you use the `DISTINCT`, `UNIQUE`, or `ALL` parameter, separate the parameter and `expr` with a space.
> - If you specify the `DISTINCT` or `UNIQUE` keyword, `order_by_clause` and `windowing_clause` cannot appear in `analytic_clause`.

Return type

The return type is the same as the data type of the `expr` parameter.

Examples

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees(manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(300, 'Wei',  '2019-09-11',23600);
INSERT INTO employees VALUES(200, 'Red', '2019-11-05', 23800);
INSERT INTO employees VALUES(100, 'Part',  '2018-10-01',24000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',23500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 23000);
INSERT INTO employees VALUES(200, 'Part',  '2018-06-11',24500);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
COMMIT;
```

Examples of the analytic function

Execute the following statement to query the number of rows in the table:

```
SELECT last_name, salary,COUNT(*) OVER (ORDER BY salary RANGE BETWEEN 50 PRECEDING
AND 150 FOLLOWING) AS mov_count FROM employees ORDER BY salary, last_name;
```

The following query result is returned:

```
+-----------+--------+-----------+
| LAST_NAME | SALARY | MOV_COUNT |
+-----------+--------+-----------+
| Errazuriz |   1400 |         1 |
| De Haan   |  11000 |         1 |
| Hartstein |  14000 |         1 |
| Bell      |  23000 |         1 |
| Ross      |  23500 |         2 |
| Wei       |  23600 |         1 |
| Red       |  23800 |         1 |
| Part      |  24000 |         1 |
| Part      |  24500 |         1 |
+-----------+--------+-----------+
```

Examples of the aggregate function

To create a table named **a** and insert data into the table, execute the following statements:

```
CREATE TABLE a (
b INT
);
INSERT INTO a VALUES (1);
INSERT INTO a VALUES (null);
INSERT INTO a VALUES (null);
INSERT INTO a VALUES (1);
INSERT INTO a VALUES (null);
INSERT INTO a VALUES (1);
INSERT INTO a VALUES (1);
```

To return the number of rows whose values are not NULL in table **a**, execute the following statement:

```
SELECT COUNT(b) FROM a;
```

The following query result is returned:

```
+----------+
| COUNT(B) |
+----------+
|        4 |
+----------+
```

To specify `COUNT(*)` to return the number of all the rows, execute the following statement:

```
SELECT COUNT(*) FROM a;
```

The following query result is returned:

```
+----------+
| COUNT(*) |
+----------+
|        7 |
+----------+
```

17.1.5.6.3.3. SUM

The `SUM` function returns the sum of values in a column that is specified by a parameter. This function uses the numeric or non-numeric data types that can be implicitly converted to numeric data types as parameters. The function returns the data type that is the same as the numeric data type of the parameter.

Syntax

```
SUM([ DISTINCT | UNQIUE | ALL ] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
DISTINCT	Removes duplicate rows and ignores the rows whose values are NULL.
UNQIUE	Removes duplicate rows and ignores the rows whose values are NULL.
ALL	Returns all the values, including duplicate rows, and ignores the rows whose values are NULL.
expr	The data column or expression of the numeric type, character type, date type, or other types.
OVER	Uses the `OVER` clause to define a window for calculation.

> **Notice**
>
> :
>
> If you specify the `DISTINCT` or `UNIQUE` keyword, `order_by_clause` and `windowing_clause` cannot appear in `analytic_clause`.

Return type

The value of the data type same as the data type of `expr` is returned.

Examples

Examples of the analytic function

To create the **employees** table and insert data into the table, execute the following statements:

```
CREATE TABLE employees(manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(300, 'Wei',  '2019-09-11',23600);
INSERT INTO employees VALUES(200, 'Red', '2019-11-05', 23800);
INSERT INTO employees VALUES(100, 'Part',  '2018-10-01',24000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',23500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 23000);
INSERT INTO employees VALUES(200, 'Part',  '2018-06-11',24500);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
COMMIT;
```

To calculate the sum of salaries, execute the following statement:

```
SELECT manager_id, last_name, salary, SUM(salary) OVER (PARTITION BY manager_id
ORDER BY salary RANGE UNBOUNDED PRECEDING) l_csum
FROM employees ORDER BY manager_id, last_name, salary, l_csum;
```

The following query result is returned:

```
+------------+-----------+--------+--------+
| MANAGER_ID | LAST_NAME | SALARY | L_CSUM |
+------------+-----------+--------+--------+
|        100 | De Haan   |  11000 |  12400 |
|        100 | Errazuriz |   1400 |   1400 |
|        100 | Hartstein |  14000 |  26400 |
|        100 | Part      |  24000 |  50400 |
|        200 | Bell      |  23000 |  23000 |
|        200 | Part      |  24500 |  94800 |
|        200 | Red       |  23800 |  70300 |
|        200 | Ross      |  23500 |  46500 |
|        300 | Wei       |  23600 |  23600 |
+------------+-----------+--------+--------+
```

Examples of the aggregate function

To calculate the sum of salaries, execute the following statement:

```
SELECT SUM(salary) FROM employees;
```

The following query result is returned:

```
+-------------+
| SUM(SALARY) |
+-------------+
|      168800 |
+-------------+
```

17.1.5.6.3.4. MAX

The `MAX` function returns the maximum value in the column that is specified by a parameter.

Syntax

```
MAX([ DISTINCT | UNIQUE | ALL ] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
DISTINCT	Removes duplicate rows from returned rows and ignores the rows whose values are NULL.

Parameter	Description
UNIQUE	Removes duplicate rows from returned rows and ignores the rows whose values are NULL.
ALL	Returns all the values, including duplicate rows, and ignores the rows whose values are NULL.
expr	The data column or expression of the numeric type, character type, date type, or other types.
OVER	Uses the `OVER` clause to define a window for calculation.

Return type

The value of the data type same as the data type of `expr` is returned.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees (manager_id INT, last_name varchar(50), hiredate varchar(50), SALARY INT);
INSERT INTO employees VALUES(100, 'Wei',  '2019-09-11',17000);
INSERT INTO employees VALUES(100, 'Red', '2019-11-05', 17000);
INSERT INTO employees VALUES(101, 'Part',  '2018-10-01',12008);
INSERT INTO employees VALUES(102, 'Wei',  '2019-09-11',9000);
INSERT INTO employees VALUES(103, 'Red', '2019-11-05', 6000);
INSERT INTO employees VALUES(104, 'Part',  '2018-10-01',8000);
COMMIT;
```

Execute the following statement to query the maximum value in the **SALARY** column:

```
SELECT manager_id, last_name, salary FROM (SELECT manager_id, last_name, salary,
MAX(salary) OVER (PARTITION BY manager_id) AS rmax_sal
FROM employees) WHERE salary = rmax_sal ORDER BY manager_id, last_name, salary;
```

The following query result is returned:

```
+------------+-----------+--------+
| MANAGER_ID | LAST_NAME | SALARY |
+------------+-----------+--------+
|        100 | Red       |  17000 |
|        100 | Wei       |  17000 |
|        101 | Part      |  12008 |
|        102 | Wei       |   9000 |
|        103 | Red       |   6000 |
|        104 | Part      |   8000 |
+------------+-----------+--------+
```

Examples of the aggregate function

Execute the following statement to query the maximum value in the **SALARY** column:

```
SELECT MAX(salary) FROM employees;
```

The following query result is returned:

```
+-------------+
| MAX(SALARY) |
+-------------+
|       17000 |
+-------------+
```

17.1.5.6.3.5. MIN

The `MIN` function returns the minimum value of a specified column.

Syntax

```
MIN([ DISTINCT | UNIQUE | ALL ] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
DISTINCT	Removes duplicate rows from returned rows and ignores the rows whose values are NULL.
UNIQUE	Removes duplicate rows from returned rows and ignores the rows whose values are NULL.
ALL	Returns all the values, including duplicate rows, and ignores the rows whose values are NULL.
expr	The data column or expression of the numeric type, character type, date type, or other types.
OVER	Uses the OVER clause to define a window for calculation.

Return type

The value of the data type same as the data type of `expr` is returned.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees (manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(100, 'Partners',  '2018-12-01',14000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',13500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 13000);
INSERT INTO employees VALUES(200, 'Part',  '2018-08-11',14000);
```

Execute the following statement to query the minimum value of the **SALARY** column:

```
SELECT manager_id, last_name, hiredate, salary, MIN(salary) OVER(PARTITION BY manager_id
ORDER BY hiredate RANGE UNBOUNDED PRECEDING) AS p_cmin
FROM employees ORDER BY manager_id, last_name, hiredate, salary;
COMMIT;
```

The following query result is returned:

```
+------------+-----------+------------+--------+--------+
| MANAGER_ID | LAST_NAME | HIREDATE   | SALARY | P_CMIN |
+------------+-----------+------------+--------+--------+
|        100 | De Haan   | 2018-05-01 |  11000 |   1400 |
|        100 | Errazuriz | 2017-07-21 |   1400 |   1400 |
|        100 | Hartstein | 2019-05-01 |  14000 |   1400 |
|        100 | Partners  | 2018-12-01 |  14000 |   1400 |
|        100 | Raphaely  | 2017-07-01 |   1700 |   1700 |
|        100 | Raphaely  | 2017-07-22 |   1700 |   1400 |
|        100 | Russell   | 2019-10-05 |  13000 |   1400 |
|        100 | Weiss     | 2019-07-11 |  13500 |   1400 |
|        200 | Bell      | 2019-05-25 |  13000 |  13000 |
|        200 | Part      | 2018-08-11 |  14000 |  14000 |
|        200 | Ross      | 2019-06-11 |  13500 |  13000 |
+------------+-----------+------------+--------+--------+
```

Examples of the aggregate function

Execute the following statement to query the minimum value of the **SALARY** column:

```
SELECT MIN(salary)  FROM employees ;
```

The following query result is returned:

```
+-------------+
| MIN(SALARY) |
+-------------+
|        1400 |
+-------------+
```

17.1.5.6.3.6. LISTAGG

The `LISTAGG` function converts columns to rows. `LISTAGG` sorts the data in each group that is specified in the `ORDER BY` clause and merges the values of the measure column. When `LISTAGG` serves as a single-set aggregate function, it performs operations on all the rows and returns a single output row. When `LISTAGG` serves as a group-set aggregate, it performs operations on each group that is defined by the `GROUP BY` clause and returns an output row for each group. When `LISTAGG` serves as an analytic function, it divides the query result set into groups based on one or more expressions in `query_partition_clause`.

Syntax

```
LISTAGG(measure_expr [,'delimiter'])WITHIN GROUP (order_by_clause)
[OVER query_partition_clause]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
OVER	Uses the `OVER` clause to define a window for calculation.
measure_expr	The value can be an expression. Null values in the measure column are ignored.
delimiter	The string that is used to separate the measure values. This clause is optional. Default value: NULL.

Return type

If the data type of the measure column is `RAW`, the returned data type is `RAW`. Otherwise, the return value is of the `VARCHAR2` type.

Examples

Examples of the analytic function

To create the **employees** table and insert data into the table, execute the following statements:

```
CREATE TABLE employees (department_id INT,manager_id INT,last_name varchar(50),hiredate varchar(50),S
ALARY INT);
INSERT INTO employees VALUES(30, 100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(30, 100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(40, 100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(50, 100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(50, 100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(70, 100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(90, 100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(90,100, 'Partners',  '2018-12-01',14000);
```

Query the employees that were hired before October 10, 2019, the departments and hire dates of the employees, and the other employees in the departments. Execute the following statement:

```
SELECT department_id "Dept", hiredate "Date", last_name "Name",LISTAGG(last_name, '; ') WITHIN GROUP
(ORDER BY hiredate, last_name) OVER (PARTITION BY department_id) as "Emp_list"
FROM employees WHERE hiredate < '2019-10-10' ORDER BY "Dept", "Date", "Name";
```

The following query result is returned:

```
+------+------------+-----------+--------------------+
| Dept | Date       | Name      | Emp_list           |
+------+------------+-----------+--------------------+
|   30 | 2017-07-01 | Raphaely  | Raphaely; De Haan  |
|   30 | 2018-05-01 | De Haan   | Raphaely; De Haan  |
|   40 | 2017-07-21 | Errazuriz | Errazuriz          |
|   50 | 2017-07-22 | Raphaely  | Raphaely; Hartstein |
|   50 | 2019-05-01 | Hartstein | Raphaely; Hartstein |
|   70 | 2019-07-11 | Weiss     | Weiss              |
|   90 | 2018-12-01 | Partners  | Partners; Russell  |
|   90 | 2019-10-05 | Russell   | Partners; Russell  |
+------+------------+-----------+--------------------+
```

Examples of the aggregate function

To create the **employees** table and insert data into the table, execute the following statements:

```
CREATE TABLE employees (department_id INT,manager_id INT,last_name varchar(50),hiredate varchar(50),S
ALARY INT);
INSERT INTO employees VALUES(30, 100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(30, 100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(30, 100, 'Errazuriz', '2017-07-01', 1400);
INSERT INTO employees VALUES(30, 100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(30, 100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(30, 100, 'Weiss',  '2019-07-01',13500);
INSERT INTO employees VALUES(30, 100, 'Russell', '2019-07-01', 13000);
INSERT INTO employees VALUES(30,100, 'Partners',  '2018-12-01',14000);
```

To query all the employees in the thirtieth department and sort the employees by hire date and last name, execute the following statement:

```
SELECT LISTAGG(last_name, '; ') WITHIN GROUP (ORDER BY hiredate, last_name)  as "Emp_list",
MIN(hiredate) as "Earliest" FROM employees WHERE department_id = 30;
```

The following query result is returned:

```
+-------------------------------------------------------------------------------+------------+
| Emp_list                                                                      | Earliest   |
+-------------------------------------------------------------------------------+------------+
| Errazuriz; Raphaely; Raphaely; De Haan; Partners; Hartstein; Russell; Weiss | 2017-07-01 |
+-------------------------------------------------------------------------------+------------+
```

17.1.5.6.3.7. ROLLUP

The `ROLLUP` function is an aggregate function. It is a simple extension to the `GROUP BY` statement. When data statistics and reports are being generated, the function returns a subtotal for each group and a grand total for all the groups. This function is more efficient than the combination of `GROUP BY` and `UNION` .

The `ROLLUP` function runs in a simple way. The function runs in the following sequence:

- Group data from right to left in descending order based on the column that is specified by the parameter.
- Calculate a subtotal for each group, and then calculate a grand total for all the groups.
- Sort the data by `ORDER BY col1 (,col2,col3,col4 ...)` .

If the number of parameters in `ROLLUP` is `N` , the result of this function is equivalent to `UNION` of `N+1` `GROUP BY` groups.

The `ROLLUP` function makes tasks that involve group statistics efficient. For example, to calculate subtotals along a hierarchical dimension, such as time or geography, you need only to use `ROLLUP(y, m, day)` or `ROLLUP(country, state, city)` for the query. Data warehouse administrators can simplify and speed up the maintenance of aggregate tables by using the `ROLLUP` function.

Syntax

```
SELECT ... GROUP BY ROLLUP(col1 [,col2...])
```

Parameters

Parameter	Description
col1	The name of the column by which data is grouped. The number of columns refers to the number of rows in the database.

Examples

To create the **group_test** table and insert data into the table, execute the following statements:

```
CREATE TABLE group_test (group_id int, job varchar2(10), name varchar2(10), salary int);
INSERT INTO group_test VALUES (10, 'Coding', 'Bruce', 1000);
INSERT INTO group_test VALUES (10, 'Programmer', 'Clair', 1000);
INSERT INTO group_test VALUES (20, 'Coding', 'Jason', 2000);
INSERT INTO group_test VALUES (20, 'Programmer', 'Joey', 2000);
INSERT INTO group_test VALUES (30, 'Coding', 'Rebecca', 3000);
INSERT INTO group_test VALUES (30, 'Programmer', 'Rex', 3000);
INSERT INTO group_test VALUES (40, 'Coding', 'Samuel', 4000);
INSERT INTO group_test VALUES (40, 'Programmer', 'Susy', 4000);
COMMIT;
```

To use `GROUP BY` to group data by **group_id**, execute the following statement:

```
SELECT group_id, SUM(salary) FROM group_test GROUP BY group_id;
```

The following query result is returned:

```
+----------+-------------+
| GROUP_ID | SUM(SALARY) |
+----------+-------------+
|       10 |        2000 |
|       20 |        4000 |
|       30 |        6000 |
|       40 |        8000 |
+----------+-------------+
```

To use the `ROLLUP` function to group data by **group_id** and calculate a grand total, execute the following statement:

```
SELECT group_id, SUM(salary) FROM group_test GROUP BY ROLLUP (group_id);
```

The following query result is returned:

```
+----------+-------------+
| GROUP_ID | SUM(SALARY) |
+----------+-------------+
|       10 |        2000 |
|       20 |        4000 |
|       30 |        6000 |
|       40 |        8000 |
|     NULL |       20000 |
+----------+-------------+
```

To use the `ROLLUP` function to group data by the **group_id** and **job** columns and calculate a grand total, execute the following statement:

```
SELECT group_id, job, SUM(salary) FROM group_test GROUP BY ROLLUP (group_id, job);
```

The following query result is returned:

```
+----------+------------+-------------+
| GROUP_ID | JOB        | SUM(SALARY) |
+----------+------------+-------------+
|       10 | Coding     |        1000 |
|       10 | Programmer |        1000 |
|       10 | NULL       |        2000 |
|       20 | Coding     |        2000 |
|       20 | Programmer |        2000 |
|       20 | NULL       |        4000 |
|       30 | Coding     |        3000 |
|       30 | Programmer |        3000 |
|       30 | NULL       |        6000 |
|       40 | Coding     |        4000 |
|       40 | Programmer |        4000 |
|       40 | NULL       |        8000 |
|     NULL | NULL       |       20000 |
+----------+------------+-------------+
```

To replace the preceding SQL statement with the combination of `GROUP BY` and `UNION`, execute the following statement:

```
SELECT group_id, job, SUM(salary) FROM group_test GROUP BY group_id, job
UNION ALL
SELECT group_id, NULL, SUM(salary) FROM group_test GROUP BY group_id
UNION ALL
SELECT NULL, NULL, SUM(salary) FROM group_test ORDER BY 1, 2;
```

The following query result is returned:

```
+----------+------------+-------------+
| GROUP_ID | JOB        | SUM(SALARY) |
+----------+------------+-------------+
|       10 | Coding     |        1000 |
|       10 | Programmer |        1000 |
|       10 | NULL       |        2000 |
|       20 | Coding     |        2000 |
|       20 | Programmer |        2000 |
|       20 | NULL       |        4000 |
|       30 | Coding     |        3000 |
|       30 | Programmer |        3000 |
|       30 | NULL       |        6000 |
|       40 | Coding     |        4000 |
|       40 | Programmer |        4000 |
|       40 | NULL       |        8000 |
|     NULL | NULL       |       20000 |
+----------+------------+-------------+
```

The output result is the same as that of the `ROLLUP` function. However, the `ROLLUP` function is simpler and more efficient.

17.1.5.6.3.8. STDDEV

The `STDDEV` function calculates the population standard deviation. The `STDDEV` function uses numeric data as arguments and returns numeric data. The difference between this function and the `STDDEV_SAMP` function is that if only one row of input data is available, `STDDEV` returns 0 but `STDDEV_SAMP` returns NULL.

In ApsaraDB for OceanBase, the value of the standard deviation is the arithmetic square root of the variance that is calculated by the `VARIANCE` function.

Syntax

```
STDDEV([ DISTINCT | UNIQUE | ALL ] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
DISTINCT	Removes duplicate keywords. This indicates that the population standard deviation of unique values is calculated.
UNIQUE	Removes duplicate keywords. This indicates that the population standard deviation of unique values is calculated.
ALL	All the numeric columns.
expr	The numeric type or the types that can be converted to the numeric type.
OVER	Uses the `OVER` clause to define a window for calculation.

> **Notice**
>
> If you specify the `DISTINCT` or `UNIQUE` keyword, `order_by_clause` and `windowing_clause` cannot appear in `analytic_clause`.

Return type

The data of the `NUMBER` type is returned.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees(manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(100, 'Partners',  '2018-12-01',14000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',13500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 13000);
INSERT INTO employees VALUES(200, 'Part',  '2018-08-11',14000);
COMMIT;
```

Call the function and execute the following statement:

```
SELECT last_name, salary, STDDEV(salary) OVER (ORDER BY hiredate) "StdDev"
FROM employees WHERE manager_id = 100 ORDER BY last_name, salary, "StdDev";
```

The following query result is returned:

```
+-----------+--------+-----------------------------------------+
| LAST_NAME | SALARY | StdDev                                  |
+-----------+--------+-----------------------------------------+
| De Haan   |  11000 | 4702.127178203498995615489088200868644482 |
| Errazuriz |   1400 |  212.132034355964257320253308631454711785 |
| Hartstein |  14000 | 6340.346993658943269176828928801701088079 |
| Partners  |  14000 | 6064.899009876421676804205219406952308814 |
| Raphaely  |   1700 |                                       0 |
| Raphaely  |   1700 |  173.205080756887729352744634150587236694 |
| Russell   |  13000 | 6026.474330580265330900400184969999384459 |
| Weiss     |  13500 | 6244.311697171159907069428668980211861012 |
+-----------+--------+-----------------------------------------+
```

Examples of the aggregate function

Call the function and execute the following statement:

```
SELECT STDDEV(salary) FROM employees WHERE manager_id = 100 ;
```

The following query result is returned:

```
+-----------------------------------------+
| STDDEV(SALARY)                          |
+-----------------------------------------+
| 6026.474330580265330900400184969999384459 |
+-----------------------------------------+
```

17.1.5.6.3.9. STDDEV_POP

The `STDDEV_POP` function calculates population standard deviation. The `STDDEV_POP` function uses numeric data as parameters and returns numeric data.

> **Notice**
> The population standard deviation is the arithmetic square root of a population variance.

Syntax

```
STDDEV_POP([ALL] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
ALL	All the numeric columns.
OVER	Uses the `OVER` clause to define a window for calculation.
expr	The expression of the numeric data types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`.

Return type

The return type is the same as the data type of the `expr` parameter.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees (manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(100, 'Partners',  '2018-12-01',14000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',13500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 13000);
INSERT INTO employees VALUES(200, 'Part',  '2018-08-11',14000);
COMMIT;
```

Call the function and execute the following statement:

```
SELECT manager_id, last_name, salary, STDDEV_POP(salary) OVER (PARTITION BY manager_id) AS pop_std
FROM employees ORDER BY manager_id, last_name, salary, pop_std;
```

The following query result is returned:

```
+------------+-----------+--------+-----------------------------------------+
| MANAGER_ID | LAST_NAME | SALARY | POP_STD                                 |
+------------+-----------+--------+-----------------------------------------+
|        100 | De Haan   |  11000 | 5637.250548804798333699350384281939588505 |
|        100 | Errazuriz |   1400 | 5637.250548804798333699350384281939588505 |
|        100 | Hartstein |  14000 | 5637.250548804798333699350384281939588505 |
|        100 | Partners  |  14000 | 5637.250548804798333699350384281939588505 |
|        100 | Raphaely  |   1700 | 5637.250548804798333699350384281939588505 |
|        100 | Raphaely  |   1700 | 5637.250548804798333699350384281939588505 |
|        100 | Russell   |  13000 | 5637.250548804798333699350384281939588505 |
|        100 | Weiss     |  13500 | 5637.250548804798333699350384281939588505 |
|        200 | Bell      |  13000 |  408.248290463863016366214012450981899069 |
|        200 | Part      |  14000 |  408.248290463863016366214012450981899069 |
|        200 | Ross      |  13500 |  408.248290463863016366214012450981899069 |
+------------+-----------+--------+-----------------------------------------+
```

Examples of the aggregate function

Call the function and execute the following statement:

```
SELECT STDDEV_POP(salary) FROM employees ;
```

The following query result is returned:

```
+-----------------------------------------+
| STDDEV_POP(SALARY)                      |
+-----------------------------------------+
| 5249.950806538512715446505486136315088416 |
+-----------------------------------------+
```

17.1.5.6.3.10. STDDEV_SAMP

The `STDDEV_SAMP` function calculates the sample standard deviation. The `STDDEV_SAMP` function uses numeric data as arguments and returns numeric data. The difference between this function and the `STDDEV` function is that if only one row of input data is available, `STDDEV` returns 0 but `STDDEV_SAMP` returns NULL.

> **Note**
>
> The sample standard deviation is the square root of a sample variance.

Syntax

```
STDDEV_SAMP([ALL] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the OVER keyword.

Parameters

Parameter	Description
ALL	All the numeric columns.
expr	The expression of the numeric types (`NUMBER` , `FLOAT` , `BINARY_FLOAT` , and `BINARY_DOUBLE`) or the data types that can be converted to the numeric types.
OVER	Uses the `OVER` clause to define a window for calculation.

Return type

The return type is the same as the data type of the `expr` parameter.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees (manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(100, 'Partners',  '2018-12-01',14000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',13500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 13000);
INSERT INTO employees VALUES(200, 'Part',  '2018-08-11',14000);
COMMIT;
```

Call the function and execute the following statement:

```
SELECT manager_id, last_name, hiredate, salary,STDDEV_SAMP(salary) OVER (PARTITION BY manager_id
ORDER BY hiredate ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS cum_sdev
FROM employees ORDER BY manager_id, last_name, hiredate, salary, cum_sdev;
```

The following result is returned:

```
+------------+-----------+------------+---------+-----------------------------------------+
| MANAGER_ID | LAST_NAME | HIREDATE   | SALARY | CUM_SDEV                                 |
+------------+-----------+------------+---------+-----------------------------------------+
|        100 | De Haan   | 2018-05-01 |  11000 | 4702.127178203498995615489088200868644482 |
|        100 | Errazuriz | 2017-07-21 |   1400 |  212.13203435596425732025330863145471785 |
|        100 | Hartstein | 2019-05-01 |  14000 | 6340.346993658943269176828928801701088079 |
|        100 | Partners  | 2018-12-01 |  14000 | 6064.899009876421676804205219406952308814 |
|        100 | Raphaely  | 2017-07-01 |   1700 |                                     NULL |
|        100 | Raphaely  | 2017-07-22 |   1700 |  173.20508075688772935274463415058723669 4 |
|        100 | Russell   | 2019-10-05 |  13000 | 6026.474330580265330900400184969999384459 |
|        100 | Weiss     | 2019-07-11 |  13500 | 6244.311697171159907069428668980211861012 |
|        200 | Bell      | 2019-05-25 |  13000 |  707.106781186547524400844362104849039285 |
|        200 | Part      | 2018-08-11 |  14000 |                                     NULL |
|        200 | Ross      | 2019-06-11 |  13500 |                                      500 |
+------------+-----------+------------+---------+-----------------------------------------+
```

Examples of the aggregate function

Call the function and execute the following statement:

```
SELECT STDDEV_SAMP(salary) FROM employees ;
```

The following query result is returned:

```
+-----------------------------------------+
| STDDEV_SAMP(SALARY)                     |
+-----------------------------------------+
| 5506.194858355615640082358245403620332764 |
+-----------------------------------------+
```

17.1.5.6.3.11. VARIANCE

The `VARIANCE` function returns the variance of the column that is specified by a parameter.

Syntax

```
VARIANCE([ DISTINCT | UNIQUE | ALL ] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
DISTINCT	Removes duplicate values from the column and ignores NULL values in the column during the query.
UNIQUE	Removes duplicate values from the column and ignores NULL values in the column during the query.

Parameter	Description
ALL	Retains duplicate values in the column and ignores NULL values in the column during the query. Default value: `ALL`.
expr	The data column or expression of the numeric type, character type, date type, or other types.
OVER	Uses the `OVER` clause to define a window for calculation.

> **Notice**
>
> If you specify the `DISTINCT` or `UNIQUE` keyword, `order_by_clause` and `windowing_clause` cannot appear in `analytic_clause`.

Return type

The data of the `NUMBER` type is returned.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees (manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(100, 'Partners',  '2018-12-01',14000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',13500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 13000);
INSERT INTO employees VALUES(200, 'Part',  '2018-08-11',14000);
COMMIT;
```

Execute the following statement to calculate the variance of the **salary** column:

```
SELECT last_name, salary, VARIANCE(salary)  OVER (ORDER BY hiredate) "Variance"
FROM employees WHERE manager_id = 100 ORDER BY last_name, salary, "Variance";
```

The following query result is returned:

```
+-----------+--------+-----------------------------------------+
| LAST_NAME | SALARY | Variance                                |
+-----------+--------+-----------------------------------------+
| De Haan   |  11000 |                                22110000 |
| Errazuriz |   1400 |                                   45000 |
| Hartstein |  14000 |                                40200000 |
| Partners  |  14000 |                                36783000 |
| Raphaely  |   1700 |                                       0 |
| Raphaely  |   1700 |                                   30000 |
| Russell   |  13000 | 36318392.857142857142857142857142857142 |
| Weiss     |  13500 | 38991428.571428571428571428571428571428 |
+-----------+--------+-----------------------------------------+
```

Examples of the aggregate function

Execute the following statement to calculate the variance of the **salary** column:

```
SELECT VARIANCE(salary)  FROM employees;
```

The following query result is returned:

```
+----------------------------------------+
| VARIANCE(SALARY)                       |
+----------------------------------------+
| 30318181.818181818181818181818181818182 |
+----------------------------------------+
```

17.1.5.6.3.12. APPROX_COUNT_DISTINCT

The `APPROX_COUNT_DISTINCT` function is an aggregate function. It calculates the number of rows in a column where duplicates are removed, and can return only one approximate value. You can use this function to further calculate the selectivity of the referenced column.

Compared with the `COUNT(DISTINCT x)` function, `APPROX_COUNT_DISTINCT` returns an approximate value. Therefore, the calculation speed of APPROX_COUNT_DISTINCT is super high. It often takes a long time for `COUNT(DISTINCT x)` to process a large amount of data. `APPROX_COUNT_DISTINCT` sacrifices a small amount of accuracy for a significant increase in computational efficiency.

Syntax

```
APPROX_COUNT_DISTINCT(expr)
```

Parameters

Parameter	Description
expr	The numeric column.

Return type

The data of the `NUMBER` type is returned.

Examples

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees (manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(100, 'Partners',  '2018-12-01',14000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',13500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 13000);
INSERT INTO employees VALUES(200, 'Part',  '2018-08-11',14000);
COMMIT;
```

Execute the following statement:

```
SELECT last_name, salary, APPROX_COUNT_DISTINCT(salary) OVER (ORDER BY hiredate) "Variance"
FROM employees WHERE manager_id = 100 ORDER BY last_name, salary, "Variance";
```

The following query result is returned:

```
+-----------+--------+----------+
| LAST_NAME | SALARY | Variance |
+-----------+--------+----------+
| De Haan   |  11000 |        3 |
| Errazuriz |   1400 |        2 |
| Hartstein |  14000 |        4 |
| Partners  |  14000 |        4 |
| Raphaely  |   1700 |        1 |
| Raphaely  |   1700 |        2 |
| Russell   |  13000 |        6 |
| Weiss     |  13500 |        5 |
+-----------+--------+----------+
```

17.1.5.6.4. Analytic functions

17.1.5.6.4.1. Description of window functions

Analytic functions (also called window functions) and aggregate functions perform aggregate operations on a group of rows (a set of rows). The difference is that aggregate functions return a value (a row) for each group, but window functions return multiple values (multiple rows) for each group. A group of rows is also known as a window and is defined by `analytic_clause`. Window sizes depend on the actual number of rows or a logical interval, such as time.

To trigger an analytic function, you must use a special keyword `OVER` to specify a window. A window consists of three parts:

- Partitioning specifications: used to split input rows into different partitions. This process is similar to the splitting process of the `GROUP BY` clause.

- Sorting specifications: used to determine the order in which input data rows are executed in the window function.
- Window boundary: specifies a window boundary for calculating data. The default value is `RANGE UNBOUNDED PRECEDING`. This boundary contains all data in all the rows that range from the start row to the current row in the current partition.

Analytic functions are the last group of operations that are performed in a query except for the final `ORDER BY` clause. Before window functions are processed, all the `JOIN` operations and `WHERE`, `GROUP BY`, and `HAVING` clauses must be completed. Therefore, window functions can appear in only the select list or the `ORDER BY` clause.

Analytic functions are generally used to calculate cumulative, moving, centered, and reporting aggregates.

Syntax

analytic_function

```
analytic_function([ arguments ]) OVER (analytic_clause)
```

analytic_clause

```
[ query_partition_clause ] [ order_by_clause [ windowing_clause ] ]
```

query_partition_clause

```
PARTITION BY { expr[, expr ]... | ( expr[, expr ]... ) }
```

order_by_clause

```
ORDER [ SIBLINGS ] BY{ expr | position | c_alias } [ ASC | DESC ] [ NULLS FIRST | NULLS LAST ] [, { e
xpr | position | c_alias } [ ASC | DESC ][ NULLS FIRST | NULLS LAST ]]...
```

windowing_clause

```
{ ROWS | RANGE } { BETWEEN { UNBOUNDED PRECEDING | CURRENT ROW | value_expr { PRECEDING | FOLLOWING }
} AND{ UNBOUNDED FOLLOWING | CURRENT ROW | value_expr { PRECEDING | FOLLOWING } } | { UNBOUNDED PRECE
DING | CURRENT ROW| value_expr PRECEDING}}
```

The following sections describe the semantics of the syntax.

analytic_function

analytic_function specifies the name of an analytic function.

arguments

The parameters (arguments). Analytic functions use 0 to 3 parameters. The parameters can be numeric data types or non-numeric data types that can be implicitly converted to numeric data types. ApsaraDB for OceanBase determines the parameter that has the highest numeric precedence based on the precedence of data types. Then, it implicitly converts the remaining parameters to the data type of the parameter that has the highest numeric precedence. The return type is also the data type of the parameter that has the highest numeric precedence, unless otherwise noted for a single function.

analytic_clause

The analytic clause (analytic_clause). Use `OVER analytic_clause` to indicate that the function performs operations on a query result set. This clause is calculated after the `FROM`, `WHERE`, `GROUP BY`, and `HAVING` clauses. You can use this clause to specify analytic functions in the select list or the ORDER BY clause. If you need to filter the results of a query based on an analytic function, nest these functions in the parent query and filter the results of the nested subquery.

> **Notice**
>
> - You cannot nest analytic functions by specifying analytic functions in analytic_clause. However, you can specify an analytic function in a subquery and calculate another analytic function over the subquery.
> - You can use user-defined analytic functions and built-in analytic functions to specify `analytic_clause`.

query_partition_clause

The partitioning clause (query_partition_clause). Use the `PARTITION BY` clause to partition a query result set to groups based on one or more `value_expr`. If you omit this clause, the function considers all the rows in the query result set as a single group.

You can specify multiple analytic functions in the same query. Each function has the same or different PARTITION BY keys. If you use `query_partition_clause` to specify an analytic function and the queried objects have the parallel attribute, the function calculations are also parallelized.

Valid values of `value_expr` are constants, columns, non-analytic functions, function expressions, or expressions that involve one of them.

order_by_clause

Use the sorting clause `order_by_clause` to specify how data is sorted in a partition. For all the analytic functions, you can sort values in a partition on multiple keys. Each key is defined by `value_expr` and is qualified by a sorting sequence.

In each function, you can specify multiple sorting expressions. This is especially useful when you use the functions that sort values.

When `order_by_clause` generates the identical values for multiple rows, the function has the following behavior:

- `CUME_DIST`, `DENSE_RANK`, `NTILE`, `PERCENT_RANK`, and `RANK` return the same result for each row.
- `ROW_NUMBER` assigns a distinct value to each row even if a value that is based on `order_by_clause` is available. The value is based on the order in which the row is processed. This order may be nondeterministic if `ORDER BY` cannot implement a total sorting.
- For other analytic functions, the result depends on the window rules. If you specify a logical window that has the `RANGE` keyword, the function returns the same result for each row. If you use the `ROWS` keyword to specify a physical window, the result is nondeterministic.

Limits on the ORDER BY clause

The `ORDER BY` clause is subject to the following limits:

- In analytic functions, `order_by_clause` must use an expression (expr). The `SIBLINGS` keyword is invalid. This keyword is relevant in only hierarchical queries. Position (position) and column aliases (c_alias) are also invalid. Otherwise, this `order_by_clause` clause is the same as the sorting command of an overall query or a subquery.
- An analytic function that uses the RANGE keyword can use multiple sort keys in the `ORDER BY` clause of this function. You must specify the following windows:
  - `RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW`, `RANGE UNBOUNDED PRECEDING` for short
  - `RANGE BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING`
  - `RANGE BETWEEN CURRENT ROW AND CURRENT ROW`
  - `RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING`

Window boundaries other than the preceding four windows can have only one sort key in the `ORDER BY` clause of the analytic function. This limit does not apply to the window boundaries that are specified by the `ROW` keyword.

ASC or DESC keyword

Specify the sorting sequence. `ASC` refers to the ascending order and `DESC` refers to the descending order. The default value is the ascending order (ASC).

NULLS FIRST or NULLS LAST keyword

`nulls first` and `nulls last` in `order_by_clause`. `nulls first` indicates that NULL values are processed as minimum values during sorting. `nulls last` indicates that NULL values are processed as maximum values during sorting.

windowing_clause

The window function clause (windowing_clause). You can use `windowing_clause` in some analytic functions. The following keywords are relevant:

ROWS or RANGE keyword

These keywords define a window for calculating the function result for each row. Then, the function is applied to all the rows in the window. The window moves through a query result set or a partition from top to bottom. A window is also called `FRAME`. ApsaraDB for OceanBase supports the following window statements:

- `ROWS`: specifies the window in physical units (rows).
- `RANGE`: specifies the window as a logical offset. The default method is `RANGE UNBOUNDED PRECEDING`. You can use window functions in analytic functions. To use `windowing_clause`, you must add `order_by_clause`.

If the window boundaries are defined by the `RANGE` clause in `windowing_clause`, you can specify only one expression in `order_by_clause`. For more information, see the limits on the ORDER BY clause. The value returned by an analytic function that has a logical offset is always deterministic. However, the value returned by an analytic function that has a physical offset may generate nondeterministic results. An analytic function that has a physical offset can return a deterministic value only when the sorting expression returns a unique sorting. Therefore, you must specify multiple columns in `order_by_clause` to implement the unique sorting.

BETWEEN ... AND keyword

Use the `BETWEEN ... AND` clause to specify the start point and end point of the window. The first expression (before AND) defines the start point. The second expression (after AND) defines the end point. If you omit `BETWEEN` and specify only one end point, ApsaraDB for OceanBase considers this point as the start point and the current row as the default end point.

UNBOUNDED PRECEDING keyword

`UNBOUNDED PRECEDING` indicates that the window starts at the first row of the partition. This is the start point instead of the end point.

UNBOUNDED FOLLOWING keyword

`UNBOUNDED FOLLOWING` indicates that the window ends at the last row of the partition. This is the end point instead of the start point.

CURRENT ROW keyword

If `CURRENT ROW` serves as a start point, it specifies that the window starts from the current row or the current value. This depends on whether you have specified `ROW` or `RANGE`. In this case, the end point cannot be `value_expr PRECEDING`. If `CURRENT ROW` serves as an end point, it specifies that the window ends at the current row or the current value. This depends on whether you have specified `ROW` or `RANGE`. In this case, the start point cannot be `value_expr FOLLOWING`.

value_expr PRECEDING or value_expr FOLLOWING keyword

- If `value_expr FOLLOWING` is the start point, the end point must be `value_expr FOLLOWING`.
- If `value_expr PRECEDING` is the end point, the start point must be `value_expr PRECEDING`.

If you need to define a logical window that is defined by a time interval in the numeric format, you may need to use conversion functions.

If you specify `ROWS`:

- `value_expr` is a physical offset. It must be a constant or an expression and must be calculated as a positive number.
- If `value_expr` is part of the start point, it must calculate the part that precedes the start point and the end point as a row.

If you specify `RANGE`:

- `value_expr` is a logical offset. It must be a constant or an expression whose result is a positive numeric value or an interval literal.
- You can specify only one expression in `order_by_clause`.
- If `value_expr` is a numeric value, `ORDER BY expr` must be the numeric or `DATE` data type.
- If `value_expr` is an interval value, `ORDER BY expr` must be the `DATE` data type. If you completely omit `windowing_clause`, the default value is `RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW`.

17.1.5.6.4.2. AVG

The `AVG` function returns the average value of a numeric column.

Syntax

```
AVG([ DISTINCT | UNIQUE | ALL ] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
DISTINCT	Removes duplicate values from the data and ignores NULL values in the data during the query.
UNIQUE	Removes duplicate values from the data and ignores NULL values in the data during the query.
ALL	Retains duplicate values in the data and ignores NULL values in the data during the query. Default value: `ALL`.
expr	The expression of the numeric type or the types that can be converted to the numeric type. The numeric type can be `NUMBER`, `FLOAT`, `BINARY_FLOAT`, or `BINARY_DOUBLE`.
OVER	Uses the `OVER` clause to define a window for calculation.

> **Notice**
>
> If you specify the `DISTINCT` or `UNIQUE` keyword, `order_by_clause` and `windowing_clause` cannot appear in `analytic_clause`.

Return type

The return type is the same as the data type of the `expr` parameter.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees (manager_id INT, last_name varchar(50), hiredate varchar(50), SALARY INT);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(100, 'Partners',  '2018-12-01',14000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',13500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 13000);
INSERT INTO employees VALUES(200, 'Part',  '2018-08-11',14000);
COMMIT;
```

Execute the following statement to calculate the average value of each column:

```
SELECT manager_id, last_name, hiredate, salary, AVG(salary) OVER (PARTITION BY manager_id
ORDER BY hiredate ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING) AS c_mavg
FROM employees ORDER BY manager_id, hiredate, salary;
```

The following result is returned:

```
+------------+-----------+------------+---------+-----------------------------------------+
| MANAGER_ID | LAST_NAME | HIREDATE   | SALARY | C_MAVG                                   |
+------------+-----------+------------+---------+-----------------------------------------+
|        100 | Errazuriz | 2017-07-21 |   1400 |                                   1550 |
|        100 | Raphaely  | 2017-07-22 |   1700 |                                   4700 |
|        100 | De Haan   | 2018-05-01 |  11000 |                                   8900 |
|        100 | Partners  | 2018-12-01 |  14000 |                                  13000 |
|        100 | Hartstein | 2019-05-01 |  14000 |                                13833.333 |
|        100 | Weiss     | 2019-07-11 |  13500 |                                  13500 |
|        100 | Russell   | 2019-10-05 |  13000 |                                  13250 |
|        200 | Part      | 2018-08-11 |  14000 |                                  13500 |
|        200 | Bell      | 2019-05-25 |  13000 |                                  13500 |
|        200 | Ross      | 2019-06-11 |  13500 |                                  13250 |
+------------+-----------+------------+---------+-----------------------------------------+
```

Examples of the aggregate function

Execute the following statement to calculate the average value of **salary**:

```
SELECT AVG(salary) FROM employees;
```

The following query result is returned:

```
+-----------------------------------------+
| AVG(SALARY)                             |
+-----------------------------------------+
| 10072.727272727272727272727272727273 |
+-----------------------------------------+
```

17.1.5.6.4.3. COUNT

The `COUNT` function queries the number of rows for `expr`.

Syntax

```
COUNT({ * | [ DISTINCT | UNIQUE | ALL ] expr }) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
*	All the rows that meet the conditions and include the rows whose values are NULL.
DISTINCT	Removes duplicate rows from returned rows and ignores the rows whose values are NULL.
UNIQUE	Removes duplicate rows from returned rows and ignores the rows whose values are NULL.
ALL	Returns all the values, including duplicate rows, and ignores the rows whose values are NULL.
expr	The expression of the numeric type or the types that can be converted to the numeric type. The numeric type can be `NUMBER`, `FLOAT`, `BINARY_FLOAT`, or `BINARY_DOUBLE`.
OVER	Uses the OVER clause to define a window for calculation.

> **Notice**
>
> - The `COUNT` function never returns `NULL`. If you specify `expr`, the function returns the number of rows where `expr` is not `NULL`. If you specify `COUNT(*)`, the function returns the number of all the rows. If you use the `DISTINCT`, `UNIQUE`, or `ALL` parameter, separate the parameter and `expr` with a space.
> - If you specify the `DISTINCT` or `UNIQUE` keyword, `order_by_clause` and `windowing_clause` cannot appear in `analytic_clause`.

Return type

The return type is the same as the data type of the `expr` parameter.

Examples

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees(manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(300, 'Wei',  '2019-09-11',23600);
INSERT INTO employees VALUES(200, 'Red', '2019-11-05', 23800);
INSERT INTO employees VALUES(100, 'Part',  '2018-10-01',24000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',23500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 23000);
INSERT INTO employees VALUES(200, 'Part',  '2018-06-11',24500);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
COMMIT;
```

Examples of the analytic function

Execute the following statement to query the number of rows in the table:

```
SELECT last_name, salary,COUNT(*) OVER (ORDER BY salary RANGE BETWEEN 50 PRECEDING
AND 150 FOLLOWING) AS mov_count FROM employees ORDER BY salary, last_name;
```

The following query result is returned:

```
+-----------+--------+-----------+
| LAST_NAME | SALARY | MOV_COUNT |
+-----------+--------+-----------+
| Errazuriz |   1400 |         1 |
| De Haan   |  11000 |         1 |
| Hartstein |  14000 |         1 |
| Bell      |  23000 |         1 |
| Ross      |  23500 |         2 |
| Wei       |  23600 |         1 |
| Red       |  23800 |         1 |
| Part      |  24000 |         1 |
| Part      |  24500 |         1 |
+-----------+--------+-----------+
```

Examples of the aggregate function

To create a table named **a** and insert data into the table, execute the following statements:

```
CREATE TABLE a (
b INT
);
INSERT INTO a VALUES (1);
INSERT INTO a VALUES (null);
INSERT INTO a VALUES (null);
INSERT INTO a VALUES (1);
INSERT INTO a VALUES (null);
INSERT INTO a VALUES (1);
INSERT INTO a VALUES (1);
```

To return the number of rows whose values are not NULL in table **a**, execute the following statement:

```
SELECT COUNT(b) FROM a;
```

The following query result is returned:

```
+----------+
| COUNT(B) |
+----------+
|        4 |
+----------+
```

To specify `COUNT(*)` to return the number of all the rows, execute the following statement:

```
SELECT COUNT(*) FROM a;
```

The following query result is returned:

```
+----------+
| COUNT(*) |
+----------+
|        7 |
+----------+
```

17.1.5.6.4.4. SUM

The `SUM` function returns the sum of values in a column that is specified by a parameter. This function uses the numeric or non-numeric data types that can be implicitly converted to numeric data types as parameters. The function returns the data type that is the same as the numeric data type of the parameter.

Syntax

```
SUM([ DISTINCT | UNQIUE | ALL ] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
DISTINCT	Removes duplicate rows and ignores the rows whose values are NULL.
UNQIUE	Removes duplicate rows and ignores the rows whose values are NULL.
ALL	Returns all the values, including duplicate rows, and ignores the rows whose values are NULL.
expr	The data column or expression of the numeric type, character type, date type, or other types.
OVER	Uses the `OVER` clause to define a window for calculation.

> **Notice**
>
> :
>
> If you specify the `DISTINCT` or `UNIQUE` keyword, `order_by_clause` and `windowing_clause` cannot appear in `analytic_clause`.

Return type

The value of the data type same as the data type of `expr` is returned.

Examples

Examples of the analytic function

To create the **employees** table and insert data into the table, execute the following statements:

```
CREATE TABLE employees(manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(300, 'Wei',  '2019-09-11',23600);
INSERT INTO employees VALUES(200, 'Red', '2019-11-05', 23800);
INSERT INTO employees VALUES(100, 'Part',  '2018-10-01',24000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',23500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 23000);
INSERT INTO employees VALUES(200, 'Part',  '2018-06-11',24500);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
COMMIT;
```

To calculate the sum of salaries, execute the following statement:

```
SELECT manager_id, last_name, salary, SUM(salary) OVER (PARTITION BY manager_id
ORDER BY salary RANGE UNBOUNDED PRECEDING) l_csum
FROM employees ORDER BY manager_id, last_name, salary, l_csum;
```

The following query result is returned:

```
+------------+-----------+--------+--------+
| MANAGER_ID | LAST_NAME | SALARY | L_CSUM |
+------------+-----------+--------+--------+
|        100 | De Haan   |  11000 |  12400 |
|        100 | Errazuriz |   1400 |   1400 |
|        100 | Hartstein |  14000 |  26400 |
|        100 | Part      |  24000 |  50400 |
|        200 | Bell      |  23000 |  23000 |
|        200 | Part      |  24500 |  94800 |
|        200 | Red       |  23800 |  70300 |
|        200 | Ross      |  23500 |  46500 |
|        300 | Wei       |  23600 |  23600 |
+------------+-----------+--------+--------+
```

Examples of the aggregate function

To calculate the sum of salaries, execute the following statement:

```
SELECT SUM(salary) FROM employees;
```

The following query result is returned:

```
+-------------+
| SUM(SALARY) |
+-------------+
|      168800 |
+-------------+
```

17.1.5.6.4.5. MAX

The `MAX` function returns the maximum value in the column that is specified by a parameter.

Syntax

```
MAX([ DISTINCT | UNIQUE | ALL ] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
DISTINCT	Removes duplicate rows from returned rows and ignores the rows whose values are NULL.

Parameter	Description
UNIQUE	Removes duplicate rows from returned rows and ignores the rows whose values are NULL.
ALL	Returns all the values, including duplicate rows, and ignores the rows whose values are NULL.
expr	The data column or expression of the numeric type, character type, date type, or other types.
OVER	Uses the `OVER` clause to define a window for calculation.

Return type

The value of the data type same as the data type of `expr` is returned.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees (manager_id INT, last_name varchar(50), hiredate varchar(50), SALARY INT);
INSERT INTO employees VALUES(100, 'Wei',  '2019-09-11',17000);
INSERT INTO employees VALUES(100, 'Red', '2019-11-05', 17000);
INSERT INTO employees VALUES(101, 'Part',  '2018-10-01',12008);
INSERT INTO employees VALUES(102, 'Wei',  '2019-09-11',9000);
INSERT INTO employees VALUES(103, 'Red', '2019-11-05', 6000);
INSERT INTO employees VALUES(104, 'Part',  '2018-10-01',8000);
COMMIT;
```

Execute the following statement to query the maximum value in the **SALARY** column:

```
SELECT manager_id, last_name, salary FROM (SELECT manager_id, last_name, salary,
MAX(salary) OVER (PARTITION BY manager_id) AS rmax_sal
FROM employees) WHERE salary = rmax_sal ORDER BY manager_id, last_name, salary;
```

The following query result is returned:

```
+------------+-----------+--------+
| MANAGER_ID | LAST_NAME | SALARY |
+------------+-----------+--------+
|        100 | Red       |  17000 |
|        100 | Wei       |  17000 |
|        101 | Part      |  12008 |
|        102 | Wei       |   9000 |
|        103 | Red       |   6000 |
|        104 | Part      |   8000 |
+------------+-----------+--------+
```

Examples of the aggregate function

Execute the following statement to query the maximum value in the **SALARY** column:

```
SELECT MAX(salary) FROM employees;
```

The following query result is returned:

```
+-------------+
| MAX(SALARY) |
+-------------+
|       17000 |
+-------------+
```

17.1.5.6.4.6. MIN

The `MIN` function returns the minimum value of a specified column.

Syntax

```
MIN([ DISTINCT | UNIQUE | ALL ] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
DISTINCT	Removes duplicate rows from returned rows and ignores the rows whose values are NULL.
UNIQUE	Removes duplicate rows from returned rows and ignores the rows whose values are NULL.
ALL	Returns all the values, including duplicate rows, and ignores the rows whose values are NULL.
expr	The data column or expression of the numeric type, character type, date type, or other types.
OVER	Uses the OVER clause to define a window for calculation.

Return type

The value of the data type same as the data type of `expr` is returned.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees (manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(100, 'Partners',  '2018-12-01',14000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',13500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 13000);
INSERT INTO employees VALUES(200, 'Part',  '2018-08-11',14000);
```

Execute the following statement to query the minimum value of the **SALARY** column:

```
SELECT manager_id, last_name, hiredate, salary, MIN(salary) OVER(PARTITION BY manager_id
ORDER BY hiredate RANGE UNBOUNDED PRECEDING) AS p_cmin
FROM employees ORDER BY manager_id, last_name, hiredate, salary;
COMMIT;
```

The following query result is returned:

```
+------------+-----------+------------+--------+--------+
| MANAGER_ID | LAST_NAME | HIREDATE   | SALARY | P_CMIN |
+------------+-----------+------------+--------+--------+
|        100 | De Haan   | 2018-05-01 |  11000 |   1400 |
|        100 | Errazuriz | 2017-07-21 |   1400 |   1400 |
|        100 | Hartstein | 2019-05-01 |  14000 |   1400 |
|        100 | Partners  | 2018-12-01 |  14000 |   1400 |
|        100 | Raphaely  | 2017-07-01 |   1700 |   1700 |
|        100 | Raphaely  | 2017-07-22 |   1700 |   1400 |
|        100 | Russell   | 2019-10-05 |  13000 |   1400 |
|        100 | Weiss     | 2019-07-11 |  13500 |   1400 |
|        200 | Bell      | 2019-05-25 |  13000 |  13000 |
|        200 | Part      | 2018-08-11 |  14000 |  14000 |
|        200 | Ross      | 2019-06-11 |  13500 |  13000 |
+------------+-----------+------------+--------+--------+
```

Examples of the aggregate function

Execute the following statement to query the minimum value of the **SALARY** column:

```
SELECT MIN(salary)  FROM employees ;
```

The following query result is returned:

```
+-------------+
| MIN(SALARY) |
+-------------+
|        1400 |
+-------------+
```

17.1.5.6.4.7. LISTAGG

The `LISTAGG` function converts columns to rows. `LISTAGG` sorts the data in each group that is specified in the `ORDER BY` clause and merges the values of the measure column. When `LISTAGG` serves as a single-set aggregate function, it performs operations on all the rows and returns a single output row. When `LISTAGG` serves as a group-set aggregate, it performs operations on each group that is defined by the `GROUP BY` clause and returns an output row for each group. When `LISTAGG` serves as an analytic function, it divides the query result set into groups based on one or more expressions in `query_partition_clause`.

Syntax

```
LISTAGG(measure_expr [,'delimiter'])WITHIN GROUP (order_by_clause)
[OVER query_partition_clause]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
OVER	Uses the `OVER` clause to define a window for calculation.
measure_expr	The value can be an expression. Null values in the measure column are ignored.
delimiter	The string that is used to separate the measure values. This clause is optional. Default value: NULL.

Return type

If the data type of the measure column is `RAW`, the returned data type is `RAW`. Otherwise, the return value is of the `VARCHAR2` type.

Examples

Examples of the analytic function

To create the **employees** table and insert data into the table, execute the following statements:

```
CREATE TABLE employees (department_id INT,manager_id INT,last_name varchar(50),hiredate varchar(50),S
ALARY INT);
INSERT INTO employees VALUES(30, 100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(30, 100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(40, 100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(50, 100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(50, 100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(70, 100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(90, 100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(90,100, 'Partners',  '2018-12-01',14000);
```

Query the employees that were hired before October 10, 2019, the departments and hire dates of the employees, and the other employees in the departments. Execute the following statement:

```
SELECT department_id "Dept", hiredate "Date", last_name "Name",LISTAGG(last_name, '; ') WITHIN GROUP
(ORDER BY hiredate, last_name) OVER (PARTITION BY department_id) as "Emp_list"
FROM employees WHERE hiredate < '2019-10-10' ORDER BY "Dept", "Date", "Name";
```

The following query result is returned:

```
+------+------------+-----------+--------------------+
| Dept | Date       | Name      | Emp_list           |
+------+------------+-----------+--------------------+
|   30 | 2017-07-01 | Raphaely  | Raphaely; De Haan  |
|   30 | 2018-05-01 | De Haan   | Raphaely; De Haan  |
|   40 | 2017-07-21 | Errazuriz | Errazuriz          |
|   50 | 2017-07-22 | Raphaely  | Raphaely; Hartstein |
|   50 | 2019-05-01 | Hartstein | Raphaely; Hartstein |
|   70 | 2019-07-11 | Weiss     | Weiss              |
|   90 | 2018-12-01 | Partners  | Partners; Russell  |
|   90 | 2019-10-05 | Russell   | Partners; Russell  |
+------+------------+-----------+--------------------+
```

Examples of the aggregate function

To create the **employees** table and insert data into the table, execute the following statements:

```
CREATE TABLE employees (department_id INT,manager_id INT,last_name varchar(50),hiredate varchar(50),S
ALARY INT);
INSERT INTO employees VALUES(30, 100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(30, 100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(30, 100, 'Errazuriz', '2017-07-01', 1400);
INSERT INTO employees VALUES(30, 100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(30, 100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(30, 100, 'Weiss',  '2019-07-01',13500);
INSERT INTO employees VALUES(30, 100, 'Russell', '2019-07-01', 13000);
INSERT INTO employees VALUES(30,100, 'Partners',  '2018-12-01',14000);
```

To query all the employees in the thirtieth department and sort the employees by hire date and last name, execute the following statement:

```
SELECT LISTAGG(last_name, '; ') WITHIN GROUP (ORDER BY hiredate, last_name)  as "Emp_list",
MIN(hiredate) as "Earliest" FROM employees WHERE department_id = 30;
```

The following query result is returned:

```
+--------------------------------------------------------------------------------+------------+
| Emp_list                                                                       | Earliest   |
+--------------------------------------------------------------------------------+------------+
| Errazuriz; Raphaely; Raphaely; De Haan; Partners; Hartstein; Russell; Weiss | 2017-07-01 |
+--------------------------------------------------------------------------------+------------+
```

17.1.5.6.4.8. STDDEV

The `STDDEV` function calculates the population standard deviation. The `STDDEV` function uses numeric data as arguments and returns numeric data. The difference between this function and the `STDDEV_SAMP` function is that if only one row of input data is available, `STDDEV` returns 0 but `STDDEV_SAMP` returns NULL.

In ApsaraDB for OceanBase, the value of the standard deviation is the arithmetic square root of the variance that is calculated by the `VARIANCE` function.

Syntax

```
STDDEV([ DISTINCT | UNIQUE | ALL ] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
DISTINCT	Removes duplicate keywords. This indicates that the population standard deviation of unique values is calculated.
UNIQUE	Removes duplicate keywords. This indicates that the population standard deviation of unique values is calculated.
ALL	All the numeric columns.
expr	The numeric type or the types that can be converted to the numeric type.
OVER	Uses the `OVER` clause to define a window for calculation.

> **Notice**
>
> If you specify the `DISTINCT` or `UNIQUE` keyword, `order_by_clause` and `windowing_clause` cannot appear in `analytic_clause`.

Return type

The data of the `NUMBER` type is returned.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees(manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(100, 'Partners',  '2018-12-01',14000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',13500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 13000);
INSERT INTO employees VALUES(200, 'Part',  '2018-08-11',14000);
COMMIT;
```

Call the function and execute the following statement:

```
SELECT last_name, salary, STDDEV(salary) OVER (ORDER BY hiredate) "StdDev"
FROM employees WHERE manager_id = 100 ORDER BY last_name, salary, "StdDev";
```

The following query result is returned:

```
+-----------+--------+-----------------------------------------+
| LAST_NAME | SALARY | StdDev                                  |
+-----------+--------+-----------------------------------------+
| De Haan   |  11000 | 4702.127178203498995615489088200868644482 |
| Errazuriz |   1400 |  212.132034355964257320253308631454711785 |
| Hartstein |  14000 | 6340.346993658943269176828928801701088079 |
| Partners  |  14000 | 6064.899009876421676804205219406952308814 |
| Raphaely  |   1700 |                                       0 |
| Raphaely  |   1700 |  173.205080756887729352744634150587236694 |
| Russell   |  13000 | 6026.474330580265330900400184969999384459 |
| Weiss     |  13500 | 6244.311697171159907069428668980211861012 |
+-----------+--------+-----------------------------------------+
```

Examples of the aggregate function

Call the function and execute the following statement:

```
SELECT STDDEV(salary) FROM employees WHERE manager_id = 100 ;
```

The following query result is returned:

```
+-----------------------------------------+
| STDDEV(SALARY)                          |
+-----------------------------------------+
| 6026.474330580265330900400184969999384459 |
+-----------------------------------------+
```

17.1.5.6.4.9. STDDEV_POP

The `STDDEV_POP` function calculates population standard deviation. The `STDDEV_POP` function uses numeric data as parameters and returns numeric data.

> **Notice**
> The population standard deviation is the arithmetic square root of a population variance.

Syntax

```
STDDEV_POP([ALL] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
ALL	All the numeric columns.
OVER	Uses the `OVER` clause to define a window for calculation.
expr	The expression of the numeric data types: `NUMBER`, `FLOAT`, `BINARY_FLOAT`, and `BINARY_DOUBLE`.

Return type

The return type is the same as the data type of the `expr` parameter.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees (manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(100, 'Partners',  '2018-12-01',14000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',13500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 13000);
INSERT INTO employees VALUES(200, 'Part',  '2018-08-11',14000);
COMMIT;
```

Call the function and execute the following statement:

```
SELECT manager_id, last_name, salary, STDDEV_POP(salary) OVER (PARTITION BY manager_id) AS pop_std
FROM employees ORDER BY manager_id, last_name, salary, pop_std;
```

The following query result is returned:

```
+------------+-----------+--------+-----------------------------------------+
| MANAGER_ID | LAST_NAME | SALARY | POP_STD                                 |
+------------+-----------+--------+-----------------------------------------+
|        100 | De Haan   |  11000 | 5637.250548804798333699350384281939588505 |
|        100 | Errazuriz |   1400 | 5637.250548804798333699350384281939588505 |
|        100 | Hartstein |  14000 | 5637.250548804798333699350384281939588505 |
|        100 | Partners  |  14000 | 5637.250548804798333699350384281939588505 |
|        100 | Raphaely  |   1700 | 5637.250548804798333699350384281939588505 |
|        100 | Raphaely  |   1700 | 5637.250548804798333699350384281939588505 |
|        100 | Russell   |  13000 | 5637.250548804798333699350384281939588505 |
|        100 | Weiss     |  13500 | 5637.250548804798333699350384281939588505 |
|        200 | Bell      |  13000 |  408.248290463863016366214012450981899069 |
|        200 | Part      |  14000 |  408.248290463863016366214012450981899069 |
|        200 | Ross      |  13500 |  408.248290463863016366214012450981899069 |
+------------+-----------+--------+-----------------------------------------+
```

Examples of the aggregate function

Call the function and execute the following statement:

```
SELECT STDDEV_POP(salary) FROM employees ;
```

The following query result is returned:

```
+-----------------------------------------+
| STDDEV_POP(SALARY)                      |
+-----------------------------------------+
| 5249.950806538512715446505486136315088416 |
+-----------------------------------------+
```

17.1.5.6.4.10. STDDEV_SAMP

The `STDDEV_SAMP` function calculates the sample standard deviation. The `STDDEV_SAMP` function uses numeric data as arguments and returns numeric data. The difference between this function and the `STDDEV` function is that if only one row of input data is available, `STDDEV` returns 0 but `STDDEV_SAMP` returns NULL.

> **Note**
> The sample standard deviation is the square root of a sample variance.

Syntax

```
STDDEV_SAMP([ALL] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the OVER keyword.

Parameters

Parameter	Description
ALL	All the numeric columns.
expr	The expression of the numeric types (`NUMBER` , `FLOAT` , `BINARY_FLOAT` , and `BINARY_DOUBLE`) or the data types that can be converted to the numeric types.
OVER	Uses the `OVER` clause to define a window for calculation.

Return type

The return type is the same as the data type of the `expr` parameter.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees (manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(100, 'Partners',  '2018-12-01',14000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',13500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 13000);
INSERT INTO employees VALUES(200, 'Part',  '2018-08-11',14000);
COMMIT;
```

Call the function and execute the following statement:

```
SELECT manager_id, last_name, hiredate, salary,STDDEV_SAMP(salary) OVER (PARTITION BY manager_id
ORDER BY hiredate ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS cum_sdev
FROM employees ORDER BY manager_id, last_name, hiredate, salary, cum_sdev;
```

The following result is returned:

```
+------------+-----------+------------+--------+-----------------------------------------+
| MANAGER_ID | LAST_NAME | HIREDATE   | SALARY | CUM_SDEV                                |
+------------+-----------+------------+--------+-----------------------------------------+
|        100 | De Haan   | 2018-05-01 |  11000 | 4702.127178203498995615489088200868644482 |
|        100 | Errazuriz | 2017-07-21 |   1400 |  212.13203435596425732025330863145471178 5 |
|        100 | Hartstein | 2019-05-01 |  14000 | 6340.346993658943269176828928801701088079 |
|        100 | Partners  | 2018-12-01 |  14000 | 6064.899009876421676804205219406952308814 |
|        100 | Raphaely  | 2017-07-01 |   1700 |                                    NULL |
|        100 | Raphaely  | 2017-07-22 |   1700 |  173.20508075688772935274463415058723669 4 |
|        100 | Russell   | 2019-10-05 |  13000 | 6026.474330580265330900400184969999384459 |
|        100 | Weiss     | 2019-07-11 |  13500 | 6244.311697171159907069428668980211861012 |
|        200 | Bell      | 2019-05-25 |  13000 |  707.106781186547524400844362104849039285 |
|        200 | Part      | 2018-08-11 |  14000 |                                    NULL |
|        200 | Ross      | 2019-06-11 |  13500 |                                     500 |
+------------+-----------+------------+--------+-----------------------------------------+
```

Examples of the aggregate function

Call the function and execute the following statement:

```
SELECT STDDEV_SAMP(salary) FROM employees ;
```

The following query result is returned:

```
+-----------------------------------------+
| STDDEV_SAMP(SALARY)                     |
+-----------------------------------------+
| 5506.194858355615640082358245403620332764 |
+-----------------------------------------+
```

17.1.5.6.4.11. VARIANCE

The `VARIANCE` function returns the variance of the column that is specified by a parameter.

Syntax

```
VARIANCE([ DISTINCT | UNIQUE | ALL ] expr) [ OVER (analytic_clause) ]
```

If the function is used as an analytic function, you must use the full syntax of a window function. The function calculates a set of rows and returns multiple values. If the function is used as an aggregate function, the function aggregates a set of rows and returns only one value. In this case, you do not need to add the `OVER` keyword.

Parameters

Parameter	Description
DISTINCT	Removes duplicate values from the column and ignores NULL values in the column during the query.
UNIQUE	Removes duplicate values from the column and ignores NULL values in the column during the query.
ALL	Retains duplicate values in the column and ignores NULL values in the column during the query. Default value: `ALL`.
expr	The data column or expression of the numeric type, character type, date type, or other types.
OVER	Uses the `OVER` clause to define a window for calculation.

Notice

If you specify the `DISTINCT` or `UNIQUE` keyword, `order_by_clause` and `windowing_clause` cannot appear in `analytic_clause`.

Return type

The data of the `NUMBER` type is returned.

Examples

Examples of the analytic function

The following statements create the **employees** table and insert data into the table:

```
CREATE TABLE employees (manager_id INT,last_name varchar(50),hiredate varchar(50),SALARY INT);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-01', 1700);
INSERT INTO employees VALUES(100, 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Errazuriz', '2017-07-21', 1400);
INSERT INTO employees VALUES(100, 'Hartstein', '2019-05-01',14000);
INSERT INTO employees VALUES(100, 'Raphaely', '2017-07-22', 1700);
INSERT INTO employees VALUES(100, 'Weiss',  '2019-07-11',13500);
INSERT INTO employees VALUES(100, 'Russell', '2019-10-05', 13000);
INSERT INTO employees VALUES(100, 'Partners',  '2018-12-01',14000);
INSERT INTO employees VALUES(200, 'Ross',  '2019-06-11',13500);
INSERT INTO employees VALUES(200, 'Bell', '2019-05-25', 13000);
INSERT INTO employees VALUES(200, 'Part',  '2018-08-11',14000);
COMMIT;
```

Execute the following statement to calculate the variance of the **salary** column:

```
SELECT last_name, salary, VARIANCE(salary)  OVER (ORDER BY hiredate) "Variance"
FROM employees WHERE manager_id = 100 ORDER BY last_name, salary, "Variance";
```

The following query result is returned:

```
+-----------+--------+-----------------------------------------+
| LAST_NAME | SALARY | Variance                                |
+-----------+--------+-----------------------------------------+
| De Haan   |  11000 |                                22110000 |
| Errazuriz |   1400 |                                   45000 |
| Hartstein |  14000 |                                40200000 |
| Partners  |  14000 |                                36783000 |
| Raphaely  |   1700 |                                       0 |
| Raphaely  |   1700 |                                   30000 |
| Russell   |  13000 | 36318392.857142857142857142857142857286 |
| Weiss     |  13500 | 38991428.571428571428571428571428571857 |
+-----------+--------+-----------------------------------------+
```

Examples of the aggregate function

Execute the following statement to calculate the variance of the **salary** column:

```
SELECT VARIANCE(salary)  FROM employees;
```

The following query result is returned:

```
+----------------------------------------+
| VARIANCE(SALARY)                       |
+----------------------------------------+
| 30318181.818181818181818181818181818182 |
+----------------------------------------+
```

17.1.5.6.4.12. RANK

The `RANK` function determines the rank of a group of values based on the `ORDER BY` expression in the `OVER` clause. If the same sort values are available, the same rank is generated, and the number of rows that have the same values is recorded to the next rank.

Syntax

```
RANK()  OVER ( [ PARTITION BY expr_list ] [ ORDER BY order_list ])
```

Parameters

Parameter	Description
OVER	Uses the `OVER` clause to define a window for calculation.
PARTITION BY [col1, col2..]	Specifies the column for which the window is opened.
ORDER BY col1[asc|desc]	Specifies the value by which data is ranked.
expr_list	The numeric type or the types that can be converted to the numeric type.
order_list	Defines the data column based on which values are ranked.

Examples

To create the **course** table and insert data into the **name** and **grade** columns, execute the following statements:

```
CREATE TABLE course
(
  name VARCHAR(8),
  grade NUMBER
);
INSERT INTO course VALUES('Linda',50);
INSERT INTO course VALUES('Tan',85);
INSERT INTO course VALUES('Tom',90);
INSERT INTO course VALUES('John',95);
INSERT INTO course VALUES('Mery',55);
INSERT INTO course VALUES('Peter',60);
INSERT INTO course VALUES('Jack',65);
INSERT INTO course VALUES('Rose',70);
INSERT INTO course VALUES('Tonny',75);
INSERT INTO course VALUES('Apple',80);
COMMIT;
```

Execute the following statement:

```
SELECT name,grade ,RANK() over(ORDER BY grade DESC) FROM course;
```

The following query result is returned:

```
+-------+-------+------------------------------+
| NAME  | GRADE | RANK()OVER(ORDERBYGRADEDESC) |
+-------+-------+------------------------------+
| John  |    95 |                            1 |
| Tom   |    90 |                            2 |
| Tan   |    85 |                            3 |
| Apple |    80 |                            4 |
| Tonny |    75 |                            5 |
| Rose  |    70 |                            6 |
| Jack  |    65 |                            7 |
| Peter |    60 |                            8 |
| Mery  |    55 |                            9 |
| Linda |    50 |                           10 |
+-------+-------+------------------------------+
```

17.1.5.6.4.13. LEAD

`LEAD` is an analytic function. It provides access to multiple rows of a table without a self join. Given a series of rows that are returned from a query and a cursor position, `LEAD` provides access to a row at a physical offset beyond this position.

Syntax

```
LEAD { (value_expr [,offset [,default]]) [RESPECT|IGNORE] NULLS
|(value_expr [RESPECT|IGNORE] NULLS [,offset [,default]]) }
OVER([query_partition_clause] order_by_clause)
```

Parameters

Parameter	Description
OVER	Uses the `OVER` clause to define a window for calculation.
offset	The offset of value_expr. This parameter is optional.
default	If you do not specify the default value, the default value is `null`. If the default value is not explicitly specified in `LEAD`, the return value is `NULL`.
{RESPECT | IGNORE} NULLS	Specifies whether to ignore `NULL` values. The default value is `RESPECT NULLS` and indicates that `NULL` values are taken into consideration.

Parameter	Description
value_expr	The field to be compared. You cannot use the `LEAD` function or other analytic functions to nest `value_expr`.

> **Notice**
>
> The `LEAD` function must be followed by order_by_clause. query_partition_clause is optional.

Return type

The returned data type is not limited.

Examples

To create the **emp_msg** table and insert data into columns, execute the following statements:

```
CREATE TABLE emp_msg(deptno INT, ename VARCHAR(30),sal INT);
INSERT INTO emp_msg VALUES(20,'ADAMS',1400);
INSERT INTO emp_msg VALUES(30,'ALLEN',1900);
INSERT INTO emp_msg VALUES(30,'BLAKE',3135);
INSERT INTO emp_msg VALUES(10,'CLARK',2750);
INSERT INTO emp_msg VALUES(20,'FORD',3300);
INSERT INTO emp_msg VALUES(30,'JAMES',1250);
INSERT INTO emp_msg VALUES(20,'JONES',3275);
INSERT INTO emp_msg VALUES(10,'KING',5300);
INSERT INTO emp_msg VALUES(30,'MARTIN',1550);
INSERT INTO emp_msg VALUES(10,'MILLER',1600);
INSERT INTO emp_msg VALUES(20,'SCOTT',3300);
INSERT INTO emp_msg VALUES(20,'SWITH',1100);
INSERT INTO emp_msg VALUES(30,'TURNER',1800);
INSERT INTO emp_msg VALUES(30,'WARD',1550);
```

Query the **emp_msg** table. Replace the last five values with **Jane**. Start to append the values that are sorted by the **ename** field in ascending order from the last but five value.

```
SELECT  deptno, ename, sal, LEAD(ename,5,'Jane') OVER (ORDER BY ename) AS new_ename
FROM emp_msg;
```

The following query result is returned:

```
+--------+--------+------+-----------+
| DEPTNO | ENAME  | SAL  | NEW_ENAME |
+--------+--------+------+-----------+
|     20 | ADAMS  | 1400 | JAMES     |
|     30 | ALLEN  | 1900 | JONES     |
|     30 | BLAKE  | 3135 | KING      |
|     10 | CLARK  | 2750 | MARTIN    |
|     20 | FORD   | 3300 | MILLER    |
|     30 | JAMES  | 1250 | SCOTT     |
|     20 | JONES  | 3275 | SWITH     |
|     10 | KING   | 5300 | TURNER    |
|     30 | MARTIN | 1550 | WARD      |
|     10 | MILLER | 1600 | Jane      |
|     20 | SCOTT  | 3300 | Jane      |
|     20 | SWITH  | 1100 | Jane      |
|     30 | TURNER | 1800 | Jane      |
|     30 | WARD   | 1550 | Jane      |
+--------+--------+------+-----------+
```

17.1.5.6.4.14. LAG

`LAG` is an analytic function. It provides access to a multi-row table at the same time without a self join. Given a series of rows that are returned from a query and a cursor position, `LAG` can access a row at a given physical offset prior to the position. You can specify the offset parameter as an integer that is greater than zero. If you do not specify an offset, its default value is 1. If the offset exceeds the scope of the window, an optional value is returned. If you do not specify the default value, the default value is `NULL`.

Syntax

```
LAG { (value_expr [,offset [,default]]) [RESPECT|IGNORE] NULLS
|(value_expr [RESPECT | IGNORE] NULLS [,offset [,default] ]) }
OVER([query_partition_clause] order_by_clause)
```

Parameters

Parameter	Description
value_expr	The field to be compared. You cannot use the `LAG` function or other analytic functions to nest `value_expr`.
offset	The offset of value_expr. This parameter is optional.
default	If you do not specify the default value, the default value is `NULL`. If the default value is not explicitly specified in `LAG`, the return value is `NULL`.

Parameter	Description
{RESPECT | IGNORE} NULLS	Specifies whether to ignore `NULL` values. The default value is `RESPECT NULLS` and indicates that `NULL` values are taken into consideration.
OVER	Uses the OVER clause to define a window for calculation.

Notice

The `LAG` function must be followed by order_by_clause. query_partition_clause is optional.

Return type

The data type of the return value is not limited.

Examples

To create the **emp_msg** table and insert data into the table, execute the following statements:

```
CREATE TABLE emp_msg(deptno INT, ename varchar(30),sal INT);
INSERT INTO emp_msg VALUES(20,'ADAMS',1400);
INSERT INTO emp_msg VALUES(30,'ALLEN',1900);
INSERT INTO emp_msg VALUES(30,'BLAKE',3135);
INSERT INTO emp_msg VALUES(10,'CLARK',2750);
INSERT INTO emp_msg VALUES(20,'FORD',3300);
INSERT INTO emp_msg VALUES(30,'JAMES',1250);
INSERT INTO emp_msg VALUES(20,'JONES',3275);
INSERT INTO emp_msg VALUES(10,'KING',5300);
INSERT INTO emp_msg VALUES(30,'MARTIN',1550);
INSERT INTO emp_msg VALUES(10,'MILLER',1600);
INSERT INTO emp_msg VALUES(20,'SCOTT',3300);
INSERT INTO emp_msg VALUES(20,'SWITH',1100);
INSERT INTO emp_msg VALUES(30,'TURNER',1800);
INSERT INTO emp_msg VALUES(30,'WARD',1550);
```

Query the **emp_msg** table. Replace the last five values with **Jane**. Start to append the values that are sorted by the **ename** field in ascending order from the last but five value. Execute the following statement:

```
SELECT deptno, ename, sal, LAG(ename,5,'Jane') OVER (ORDER BY ename) AS new_ename
FROM emp_msg;
```

The following query result is returned:

```
+--------+--------+------+-----------+
| DEPTNO | ENAME  | SAL  | NEW_ENAME |
+--------+--------+------+-----------+
|     20 | ADAMS  | 1400 | Jane      |
|     30 | ALLEN  | 1900 | Jane      |
|     30 | BLAKE  | 3135 | Jane      |
|     10 | CLARK  | 2750 | Jane      |
|     20 | FORD   | 3300 | Jane      |
|     30 | JAMES  | 1250 | ADAMS     |
|     20 | JONES  | 3275 | ALLEN     |
|     10 | KING   | 5300 | BLAKE     |
|     30 | MARTIN | 1550 | CLARK     |
|     10 | MILLER | 1600 | FORD      |
|     20 | SCOTT  | 3300 | JAMES     |
|     20 | SWITH  | 1100 | JONES     |
|     30 | TURNER | 1800 | KING      |
|     30 | WARD   | 1550 | MARTIN    |
+--------+--------+------+-----------+
```

17.1.5.6.4.15. FIRST_VALUE

`FIRST_VALUE` is an analytic function. It returns the first value in a set of the ordered values. If the first value in the set is `NULL`, the function returns `NULL` unless you specify `IGNORE NULLS`. This configuration is useful for data densification.

Syntax

```
FIRST_VALUE { (expr) [ {RESPECT | IGNORE} NULLS ] | (expr [ {RESPECT | IGNORE} NULLS ]) } OVER (analytic_clause)
```

Parameters

Parameter	Description
expr	The parameter type is not limited.
OVER	Uses the `OVER` clause to define a window for calculation.
{RESPECT | IGNORE} NULLS	Specifies whether to ignore `NULL` values. The default value is `RESPECT NULLS` and indicates that `NULL` values are taken into consideration.

Parameter	Description
FROM { FIRST | LAST }	Specifies whether the calculation starts from the first or last row of the window. The default value is `FROM FIRST`. If you specify `IGNORE NULLS`, `FIRST_VALUE` returns the first non-null value in the set. If all the values are `NULL`, `NULL` is returned.

Return type

The data type is not limited.

Examples

To create the **emp_msg** table and insert data into the table, execute the following statements:

```
CREATE TABLE emp_msg(deptno INT, ename VARCHAR(30), sal INT, MGR VARCHAR(30));
INSERT INTO emp_msg VALUES(10,'CLARK', 2750, 7839);
INSERT INTO emp_msg VALUES(10,'KING', 5300, NULL);
INSERT INTO emp_msg VALUES(10,'MILLER', 1600, 7782);
INSERT INTO emp_msg VALUES(20,'ADAMS', 1400, 7788);
INSERT INTO emp_msg VALUES(20,'FORD', 3300, 7566);
INSERT INTO emp_msg VALUES(20,'JONES', 3275, 7839);
INSERT INTO emp_msg VALUES(20,'SCOTT', 3300, 7566);
INSERT INTO emp_msg VALUES(20,'SMITH', 1100, 7902);
INSERT INTO emp_msg VALUES(30,'ALLEN', 1900, 7698);
INSERT INTO emp_msg VALUES(30,'BLAKE', 3150, 7839);
INSERT INTO emp_msg VALUES(30,'JAMES', 1250, 7698);
INSERT INTO emp_msg VALUES(30,'MARTIN', 1550, 7698);
INSERT INTO emp_msg VALUES(30,'TURNER', 1800, 7698);
INSERT INTO emp_msg VALUES(30,'WARD', 1550, 7698);
```

Query the highest and the first non-null **MGR** value in the **sal** column in the **emp_msg** table and use the queried value as the **first_MGR** column.

```
SELECT deptno , ename , sal , MGR ,
FIRST_VALUE ( MGR ) IGNORE NULLS over ( ORDER BY sal DESC ROWS UNBOUNDED PRECEDING ) AS first_MGR
FROM emp_msg ORDER BY deptno , ename;
```

The following query result is returned:

```
+--------+--------+------+------+-----------+
| DEPTNO | ENAME  | SAL  | MGR  | FIRST_MGR |
+--------+--------+------+------+-----------+
|     10 | CLARK  | 2750 | 7839 | 7566      |
|     10 | KING   | 5300 | NULL | NULL      |
|     10 | MILLER | 1600 | 7782 | 7566      |
|     20 | ADAMS  | 1400 | 7788 | 7566      |
|     20 | FORD   | 3300 | 7566 | 7566      |
|     20 | JONES  | 3275 | 7839 | 7566      |
|     20 | SCOTT  | 3300 | 7566 | 7566      |
|     20 | SMITH  | 1100 | 7902 | 7566      |
|     30 | ALLEN  | 1900 | 7698 | 7566      |
|     30 | BLAKE  | 3150 | 7839 | 7566      |
|     30 | JAMES  | 1250 | 7698 | 7566      |
|     30 | MARTIN | 1550 | 7698 | 7566      |
|     30 | TURNER | 1800 | 7698 | 7566      |
|     30 | WARD   | 1550 | 7698 | 7566      |
+--------+--------+------+------+-----------+
```

17.1.5.6.4.16. LAST_VALUE

The `LAST_VALUE` function is an analytic function. It returns the last value in a set of the ordered values. If the last value in the set is `NULL`, the function returns `NULL` unless you specify `IGNORE NULLS`. This configuration is useful for data densification.

Syntax

```
LAST_VALUE {(expr) [RESPECT|IGNORE NULLS] | (expr [RESPECT|IGNORE NULLS])}
OVER (analytic_clause)
```

Parameters

Parameter	Description
OVER	Uses the `OVER` clause to define a window for calculation.
expr	You cannot use `LAST_VALUE` or other analytic functions for `expr` to nest analytic functions.
FROM { FIRST | LAST }	Specifies whether the calculation starts from the first or last row of the window. The default value is `FROM FIRST`.

Parameter	Description
{RESPECT | IGNORE} NULLS	Specifies whether to ignore `NULL` values. The default value is `RESPECT NULLS` and indicates that `NULL` values are taken into consideration. If you specify `IGNORE NULLS`, `LAST_VALUE` returns the last non-null value in the set. If all the values are `NULL`, `NULL` is returned.

Return type

The data type of the return value is not limited.

Examples

To create the **emp_msg** table and insert data into the table, execute the following statements:

```
CREATE TABLE emp_msg(deptno INT, ename varchar(30),sal INT, MGR varchar(30));
INSERT INTO emp_msg VALUES(10,'CLARK', 2750, 7839);
INSERT INTO emp_msg VALUES(10,'KING', 5300, NULL);
INSERT INTO emp_msg VALUES(10,'MILLER', 1600, 7782);
INSERT INTO emp_msg VALUES(20,'ADAMS', 1400, 7788);
INSERT INTO emp_msg VALUES(20,'FORD', 3300, 7566);
INSERT INTO emp_msg VALUES(20,'JONES', 3275, 7839);
INSERT INTO emp_msg VALUES(20,'SCOTT', 3300, 7566);
INSERT INTO emp_msg VALUES(20,'SMITH', 1100, 7902);
INSERT INTO emp_msg VALUES(30,'ALLEN', 1900, 7698);
INSERT INTO emp_msg VALUES(30,'BLAKE', 3150, 7839);
INSERT INTO emp_msg VALUES(30,'JAMES', 1250, 7698);
INSERT INTO emp_msg VALUES(30,'MARTIN', 1550, 7698);
INSERT INTO emp_msg VALUES(30,'TURNER', 1800, 7698);
INSERT INTO emp_msg VALUES(30,'WARD', 1550, 7698);
```

To query the lowest and the last non-null **MGR** value in the **sal** column in the **emp_msg** table and use the queried value as the **last_MGR** column, execute the following statement:

```
SELECT deptno , ename , sal , MGR ,
LAST_VALUE ( MGR ) IGNORE NULLS OVER (ORDER BY sal DESC ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDE
D FOLLOWING ) AS last_MGR
FROM emp_msg ORDER BY deptno , ename ;
```

The following query result is returned:

```
+--------+--------+------+------+-----------+
| DEPTNO | ENAME  | SAL  | MGR  | FIRST_MGR |
+--------+--------+------+------+-----------+
|     10 | CLARK  | 2750 | 7839 | 7839      |
|     10 | KING   | 5300 | NULL | NULL      |
|     10 | MILLER | 1600 | 7782 | 7782      |
|     20 | ADAMS  | 1400 | 7788 | 7788      |
|     20 | FORD   | 3300 | 7566 | 7566      |
|     20 | JONES  | 3275 | 7839 | 7839      |
|     20 | SCOTT  | 3300 | 7566 | 7566      |
|     20 | SMITH  | 1100 | 7902 | 7902      |
|     30 | ALLEN  | 1900 | 7698 | 7698      |
|     30 | BLAKE  | 3150 | 7839 | 7839      |
|     30 | JAMES  | 1250 | 7698 | 7698      |
|     30 | MARTIN | 1550 | 7698 | 7698      |
|     30 | TURNER | 1800 | 7698 | 7698      |
|     30 | WARD   | 1550 | 7698 | 7698      |
+--------+--------+------+------+-----------+
```

17.1.5.6.4.17. NTH_VALUE

`NTH_VALUE` returns the value of `measure_expr` in the `n` th row of the window that is defined by `analytic_clause`. The return value is of the data type of `measure_expr`.

Syntax

```
NTH_VALUE (measure_expr, n) [ FROM { FIRST | LAST } ][ { RESPECT | IGNORE } NULLS ] OVER (analytic_clause)
```

Parameters

Parameter	Description
OVER	Uses the `OVER` clause to define a window for calculation.
measure_expr	The field name.
n	n is a positive number and determines the nth row for which the measurement value is to be returned. If n is NULL, the function returns an error. If n is greater than the number of all the rows in the window, the function returns NULL.

Parameter	Description
FROM { FIRST | LAST }	Specifies whether the calculation starts from the first or last row of the window. The default value is `FROM FIRST`.
{RESPECT | IGNORE} NULLS	Specifies whether to ignore `NULL` values. The default value is `RESPECT NULLS` and indicates that `NULL` values are taken into consideration.

Return type

The data type of the return value is not limited.

Examples

To create the **emp_msg** table and insert data into the table, execute the following statements:

```
CREATE TABLE emp_msg(deptno INT, ename VARCHAR(30),sal INT, MGR VARCHAR(30), hiredate VARCHAR(50));
INSERT INTO emp_msg VALUES(10,'CLARK', 2750, 7839, '2018-05-01');
INSERT INTO emp_msg VALUES(10,'KING', 5300, NULL, '2018-05-10');
INSERT INTO emp_msg VALUES(10,'MILLER', 1600, 7782, '2018-06-01');
INSERT INTO emp_msg VALUES(20,'ADAMS', 1400, 7788, '2018-05-21');
INSERT INTO emp_msg VALUES(20,'FORD', 3300, 7566, '2018-06-01');
INSERT INTO emp_msg VALUES(20,'JONES', 3275, 7839, '2018-06-20');
INSERT INTO emp_msg VALUES(20,'SCOTT', 3300, 7566, '2018-07-01');
INSERT INTO emp_msg VALUES(20,'SMITH', 1100, 7902, '2018-07-10');
INSERT INTO emp_msg VALUES(30,'ALLEN', 1900, 7698, '2018-08-05');
INSERT INTO emp_msg VALUES(30,'BLAKE', 3150, 7839, '2018-06-10');
INSERT INTO emp_msg VALUES(30,'JAMES', 1250, 7698, '2018-09-05');
INSERT INTO emp_msg VALUES(30,'MARTIN', 1550, 7698, '2018-10-01');
INSERT INTO emp_msg VALUES(30,'TURNER', 1800, 7698, '2019-05-01');
INSERT INTO emp_msg VALUES(30,'WARD', 1550, 7698, '2019-05-10');
```

Group data by department **deptno** and query the result of the comparison between the salaries of personnel in each department and the salary amount that ranks third in this department. Execute the following statement:

```
SELECT deptno, ename, sal, nth_value(sal, 3) OVER (PARTITION BY deptno ORDER BY sal DESC
rows BETWEEN unbounded preceding AND unbounded following)  AS third_most_sal
FROM emp_msg ORDER BY deptno,sal DESC;
```

The following query result is returned:

```
+--------+--------+------+----------------+
| DEPTNO | ENAME  | SAL  | THIRD_MOST_SAL |
+--------+--------+------+----------------+
|     10 | KING   | 5300 |           1600 |
|     10 | CLARK  | 2750 |           1600 |
|     10 | MILLER | 1600 |           1600 |
|     20 | FORD   | 3300 |           3275 |
|     20 | SCOTT  | 3300 |           3275 |
|     20 | JONES  | 3275 |           3275 |
|     20 | ADAMS  | 1400 |           3275 |
|     20 | SMITH  | 1100 |           3275 |
|     30 | BLAKE  | 3150 |           1800 |
|     30 | ALLEN  | 1900 |           1800 |
|     30 | TURNER | 1800 |           1800 |
|     30 | MARTIN | 1550 |           1800 |
|     30 | WARD   | 1550 |           1800 |
|     30 | JAMES  | 1250 |           1800 |
+--------+--------+------+----------------+
```

17.1.5.6.4.18. CUME_DIST

The `CUME_DIST` function calculates the cumulative distribution of a value in a group of values. The range of the return value is `0 < CUME_DIST <= 1` Tie values always evaluate to the same cumulative distribution value. This function uses the numeric or non-numeric data types that can be implicitly converted to numeric data types as parameters. ApsaraDB for OceanBase determines the parameter that has the highest numeric precedence, implicitly converts the remaining parameters to that data type, performs calculations, and returns `NUMBER`.

`CUME_DIST` that serves as an analytic function calculates the relative position of a specified value in a group of values. Assume that the ascending order is used for row r. The cume_dist of r is obtained by dividing the number of rows whose values are less than or equal to the value of r by the number of the calculated rows (the entire query result set or a partition).

Syntax

```
CUME_DIST() OVER ([ query_partition_clause ] order_by_clause)
```

Parameters

Parameter	Description
expr	The expression of the `NUMBER` type or the types that can be implicitly converted to the `NUMBER` data type.

Parameter	Description
{RESPECT | IGNORE} NULLS	Specifies whether to ignore `NULL` values. The default value is `RESPECT NULLS` and indicates that `NULL` values are taken into consideration.

Return type

The return value is of the `NUMBER` data type.

Examples

To create the **emp_msg** table and insert data into the table, execute the following statements:

```
CREATE TABLE emp_msg(deptno INT, ename varchar(30), sal INT, MGR varchar(30));
INSERT INTO emp_msg VALUES(10,'CLARK', 2750, 7839);
INSERT INTO emp_msg VALUES(10,'KING', 5300, NULL);
INSERT INTO emp_msg VALUES(10,'MILLER', 1600, 7782);
INSERT INTO emp_msg VALUES(20,'ADAMS', 1400, 7788);
INSERT INTO emp_msg VALUES(20,'FORD', 3300, 7566);
INSERT INTO emp_msg VALUES(20,'JONES', 3275, 7839);
INSERT INTO emp_msg VALUES(20,'SCOTT', 3300, 7566);
INSERT INTO emp_msg VALUES(20,'SMITH', 1100, 7902);
INSERT INTO emp_msg VALUES(30,'ALLEN', 1900, 7698);
INSERT INTO emp_msg VALUES(30,'BLAKE', 3150, 7839);
INSERT INTO emp_msg VALUES(30,'JAMES', 1250, 7698);
INSERT INTO emp_msg VALUES(30,'MARTIN', 1550, 7698);
INSERT INTO emp_msg VALUES(30,'TURNER', 1800, 7698);
INSERT INTO emp_msg VALUES(30,'WARD', 1550, 7698);
```

To return the `CUME_DIST` value of the specified column **sal** in the same **deptno**, execute the following statement:

```
SELECT deptno , ename , sal, cume_dist ( ) over ( partition BY deptno ORDER BY sal DESC ) "RANK"
FROM emp_msg WHERE sal>2000;
```

The following query result is returned:

```
+--------+-------+------+-------------------------------------------+
| DEPTNO | ENAME | SAL  | RANK                                      |
+--------+-------+------+-------------------------------------------+
|     10 | KING  | 5300 |                                        .5 |
|     10 | CLARK | 2750 |                                         1 |
|     20 | FORD  | 3300 | .666666666666666666666666666666666666667 |
|     20 | SCOTT | 3300 | .666666666666666666666666666666666666667 |
|     20 | JONES | 3275 |                                         1 |
|     30 | BLAKE | 3150 |                                         1 |
+--------+-------+------+-------------------------------------------+
```

17.1.5.6.4.19. DENSE_RANK

`DENSE_RANK` calculates the rank of a row in an ordered group of rows and returns the rank as `NUMBER`. Ranks are consecutive integers that start from 1. The maximum rank value is the number of unique values that are returned by the query. Rank values are not skipped in the case of ties. The rows that have same values for the ranking criteria receive the same rank. This function is useful for top-N and bottom-N reporting.

When `DENSE_RANK` serves as an analytic function, it calculates the rank of each row that is returned by the query among other rows based on the value of `value_exprs` in `order_by_clause`.

Syntax

```
DENSE_RANK( ) OVER([ query_partition_clause ] order_by_clause)
```

Parameters

Parameter	Description
OVER	Uses the `OVER` clause to define a window for calculation.

Return type

The return value is of the `NUMBER` data type.

Examples

To create the **emp_msg** table and insert data into the table, execute the following statements:

```
CREATE TABLE emp_msg(deptno INT, ename varchar(30), sal INT, MGR varchar(30));
INSERT INTO emp_msg VALUES(10,'CLARK', 2750, 7839);
INSERT INTO emp_msg VALUES(10,'KING', 5300, NULL);
INSERT INTO emp_msg VALUES(10,'MILLER', 1600, 7782);
INSERT INTO emp_msg VALUES(20,'ADAMS', 1400, 7788);
INSERT INTO emp_msg VALUES(20,'FORD', 3300, 7566);
INSERT INTO emp_msg VALUES(20,'JONES', 3275, 7839);
INSERT INTO emp_msg VALUES(20,'SCOTT', 3300, 7566);
INSERT INTO emp_msg VALUES(20,'SMITH', 1100, 7902);
INSERT INTO emp_msg VALUES(30,'ALLEN', 1900, 7698);
INSERT INTO emp_msg VALUES(30,'BLAKE', 3150, 7839);
INSERT INTO emp_msg VALUES(30,'JAMES', 1250, 7698);
INSERT INTO emp_msg VALUES(30,'MARTIN', 1550, 7698);
INSERT INTO emp_msg VALUES(30,'TURNER', 1800, 7698);
INSERT INTO emp_msg VALUES(30,'WARD', 1550, 7698);
```

The following example shows the analytic function feature. Execute the following statement:

```
SELECT deptno, ename, sal, DENSE_RANK ( ) OVER ( partition BY deptno ORDER BY sal DESC ) "RANK"
FROM emp_msg WHERE sal>2000;
```

The following query result is returned:

```
+--------+-------+------+------+
| DEPTNO | ENAME | SAL  | RANK |
+--------+-------+------+------+
|     10 | KING  | 5300 |    1 |
|     10 | CLARK | 2750 |    2 |
|     20 | SCOTT | 3300 |    1 |
|     20 | FORD  | 3300 |    1 |
|     20 | JONES | 3275 |    2 |
|     30 | BLAKE | 3150 |    1 |
+--------+-------+------+------+
```

17.1.5.6.4.20. NTILE

The `NTILE` function divides an ordered dataset into a number of buckets that are indicated by `expr` and assigns an appropriate bucket number to each row. The bucket number ranges from 1 to `expr`. For each partition, the value of `expr` must be resolved to a positive constant. If `expr` is a non-integer constant, ApsaraDB for OceanBase truncates this value to an integer. The return value is `NUMBER`.

The number of rows in the buckets can differ by at most 1. Each bucket is assigned with a remainder value (the remainder of the number of rows divided by the number of buckets). This starts from bucket 1. If `expr` is greater than the number of rows, a number of buckets equal to the number of rows are filled and the remaining buckets are empty.

You cannot use `NTILE` or other analytic functions to nest analytic functions. However, you can use other built-in function expressions in `expr`.

Syntax

```
NTILE(expr) OVER ([query_partition_clause] order_by_clause)
```

Parameters

Parameter	Description
expr	The value can be only a positive constant.
OVER	Uses the `OVER` clause to define a window for calculation.

Return type

Data of the `NUMERIC` type is returned.

Examples

Categorize students to four levels based on scores to determine awards for students. To create the **course** table and insert data into the table, execute the following statements:

```
CREATE TABLE course
(
  name VARCHAR(8),
  grade NUMBER
);
INSERT INTO course VALUES('Linda',50);
INSERT INTO course VALUES('Tan',85);
INSERT INTO course VALUES('Tom',90);
INSERT INTO course VALUES('John',95);
INSERT INTO course VALUES('Mery',55);
INSERT INTO course VALUES('Peter',60);
INSERT INTO course VALUES('Jack',65);
INSERT INTO course VALUES('Rose',70);
INSERT INTO course VALUES('Tonny',75);
INSERT INTO course VALUES('Apple',80);
COMMIT;
```

Execute the following statement:

```
SELECT name, grade, ntile(4) OVER (ORDER BY grade DESC) til FROM course;
```

The following query result is returned:

```
+-------+-------+------+
| NAME  | GRADE | TIL  |
+-------+-------+------+
| John  |    95 |    1 |
| Tom   |    90 |    1 |
| Tan   |    85 |    1 |
| Apple |    80 |    2 |
| Tonny |    75 |    2 |
| Rose  |    70 |    2 |
| Jack  |    65 |    3 |
| Peter |    60 |    3 |
| Mery  |    55 |    4 |
| Linda |    50 |    4 |
+-------+-------+------+
```

17.1.5.6.4.21. PERCENT_RANK

The `PERCENT_RANK` function is similar to the `CUME_DIST` (cumulative distribution) function. The return value ranges from 0 to 1. The `PERCENT_RANK` function of the first row in a set is 0. The return value is `NUMBER`.

Syntax

```
PERCENT_RANK( ) OVER ([query_partition_clause] order_by_clause)
```

Parameters

Parameter	Description

Parameter	Description
OVER	Uses the `OVER` clause to define a window for calculation.

Return type

The numeric data type is returned.

Examples

Categorize students to four levels based on scores to determine the awards for the students. To create the **course_rank** table and insert data into the table, execute the following statements:

```
CREATE TABLE course_rank
(
  name VARCHAR(8),
    id NUMBER
);
INSERT INTO course_rank VALUES('Linda',1);
INSERT INTO course_rank VALUES('Tan',2);
INSERT INTO course_rank VALUES('Tom',3);
INSERT INTO course_rank VALUES('John',4);
INSERT INTO course_rank VALUES('Mery',5);
COMMIT;
```

Execute the following statement:

```
SELECT name, id ,percent_rank() OVER (ORDER BY id) AS pr1 FROM course_rank;
```

The following query result is returned:

```
+-------+------+------+
| NAME  | ID   | PR1  |
+-------+------+------+
| Linda |    1 |    0 |
| Tan   |    2 |  .25 |
| Tom   |    3 |   .5 |
| John  |    4 |  .75 |
| Mery  |    5 |    1 |
+-------+------+------+
```

17.1.5.6.4.22. RATIO_TO_REPORT

The `RATIO_TO_REPORT` function calculates the ratio of a value to the sum of a group of values.

Syntax

```
RATIO_TO_REPORT(expr) OVER ([query_partition_clause])
```

Parameter

Parameter	Description
expr	The value can be only a positive constant.
OVER	Uses the `OVER` clause to define a window for calculation.

Return type

The numeric data is returned.

Examples

The following example shows the ratio of the output of employees to the total output of the department. To create the **product** table and insert data into the data, execute the following statements:

```
CREATE TABLE product (name VARCHAR(8), deptno NUMBER, output NUMBER);
INSERT INTO product VALUES('Linda',100,5050);
INSERT INTO product VALUES('Tan',1001,8500);
INSERT INTO product VALUES('Tom',1001,3900);
INSERT INTO product VALUES('John',100,29500);
INSERT INTO product VALUES('Mery',1001,1500);
INSERT INTO product VALUES('Peter',100,1060);
COMMIT;
```

Execute the following statement:

```
SELECT name, OUTPUT, deptno, RATIO_TO_REPORT(output) OVER (partition BY deptno) FROM product;
```

The following query result is returned:

```
+-------+--------+--------+-----------------------------------------------+
| NAME  | OUTPUT | DEPTNO | RATIO_TO_REPORT(OUTPUT)OVER(PARTITIONBYDEPTNO) |
+-------+--------+--------+-----------------------------------------------+
| Linda |   5050 |    100 |      .141814097163718056725638865487222690255 |
| John  |  29500 |    100 |      .828418983431620331367593372648132547037 |
| Peter |   1060 |    100 |      .029766919404661611906767761864644762707 |
| Tan   |   8500 |   1001 |      .611510791366906474820143884892086330935 |
| Tom   |   3900 |   1001 |      .280575539568345323741007194244604316546 |
| Mery  |   1500 |   1001 |       .107913669064748201438848920863309352518 |
+-------+--------+--------+-----------------------------------------------+
```

17.1.5.6.4.23. ROW_NUMBER

The `ROW_NUMBER` function assigns a unique number to each row to which the function is applied, regardless of each row in the partition or each row returned by the query. The function returns values starting from 1 based on the ordered sequence of rows specified in the `order_by_clause`. You can use a subquery to nest another subquery in a query that retrieves `ROW_NUMBER` in a specified range. This way, you can find a precise subset of rows from the results of the inner query. You can use this function to implement `top-n`, `bottom-n`, and `inner-n` reporting. The query must ensure a deterministic sort order for consistent results.

Syntax

```
ROW_NUMBER( ) OVER ([ query_partition_clause ] order_by_clause)
```

Parameters

Parameter	Description
OVER	Uses the `OVER` clause to define a window for calculation.

Return type

The numeric data is returned.

Examples

Rank the employees by employee output in the unit of department. To create the **product** table and insert data into the table, execute the following statements:

```
CREATE TABLE product(name VARCHAR(8), deptno NUMBER, output NUMBER);
INSERT INTO product VALUES('Linda',100,5050);
INSERT INTO product VALUES('Tan',1001,8500);
INSERT INTO product VALUES('Tom',1001,3900);
INSERT INTO product VALUES('John',100,29500);
INSERT INTO product VALUES('Mery',1001,1500);
INSERT INTO product VALUES('Peter',100,1060);
COMMIT;
```

Execute the following statement:

```
SELECT name,OUTPUT,deptno,ROW_NUMBER() OVER (partition BY deptno ORDER BY OUTPUT DESC) FROM product;
```

The following query result is returned:

```
+-------+--------+--------+-----------------------------------------------------+
| NAME  | OUTPUT | DEPTNO | ROW_NUMBER()OVER(PARTITIONBYDEPTNOORDERBYOUTPUTDESC) |
+-------+--------+--------+-----------------------------------------------------+
| John  |  29500 |    100 |                                                   1 |
| Linda |   5050 |    100 |                                                   2 |
| Peter |   1060 |    100 |                                                   3 |
| Tan   |   8500 |   1001 |                                                   1 |
| Tom   |   3900 |   1001 |                                                   2 |
| Mery  |   1500 |   1001 |                                                   3 |
+-------+--------+--------+-----------------------------------------------------+
```

17.1.5.7. Expressions

17.1.5.7.1. Overview of SQL expressions

Expressions are used to calculate data values. An expression is a combination of one or more components, such as numeric values, operators, and SQL functions. In general, an expression assumes the data types of its components.

> **Notice**
>
> The values of the `NLS_COMP` and `NLS_SORT` parameters jointly affect the sorting and comparison of characters. If the `NLS_COMP` parameter in the database is set to `LINGUISTIC`, all the entities described in this chapter follow the rules that are specified by the `NLS_SORT` parameter. If the `NLS_COMP` parameter is not set to `LINGUISTIC`, the functions are not affected by `NLS_SORT`. You can directly specify the value of `NLS_SORT`. If you do not specify it, it inherits the value of `NLS_LANGUAGE`.

The result of the following simple expression is 4 and of the `NUMBER` data type. This data type is consistent with the data type of the components.

```
2*2
```

The following complex expression uses functions and operators. This expression adds seven days to the current date, removes the time portion, and then converts the result to the `CHAR` data type.

```
TO_CHAR(TRUNC(SYSDATE+7))
```

You can also use expressions in the following scenarios:

- The selected columns in `SELECT` statements
- The `WHERE` and `HAVING` clauses
- The `CONNECT BY`, `START WITH`, and `ORDER BY` clauses
- The `VALUES` clauses of `INSERT` statements
- The `SET` clauses of `UPDATE` statements

For example, you can use an expression to replace the string **Smith** in the `SET` clause of the following `UPDATE` statement:

```
SET last_name = 'Smith';
```

In the `SET` clause, the expression `INITCAP(last_name)` is used to replace the string **Smith**.

```
SET last_name = INITCAP(last_name);
```

In ApsaraDB for OceanBase, not all expressions can be directly used by SQL statements. For more information, see the limits on expressions in the SQL statements topic.

17.1.5.7.2. Simple expressions

A simple expression can be a column, pseudocolumn, constant, sequence number, or null value.

In addition to user schemas, `schema` can also be **PUBLIC** to specify the attribute of a synonym for a table or a view. You can use **PUBLIC** to specify a public synonym in only a DML statement. You cannot use PUBLIC in a DDL statement. `NCHAR` and `NVARCHAR2` are not valid pseudocolumn data types.

The following list provides some valid simple expressions:

- `employees.last_name`
- `'this is a text string'`
- `10`
- `N 'this is an NCHAR string'`

17.1.5.7.3. Compound expressions

A compound expression is an expression that is a combination of other types of expressions.

You can use a built-in function as an expression. However, in compound expressions, some combinations of functions are inappropriate and are discarded. For example, the `LENGTH` function is inapplicable in an aggregate function.

You can use the `PRIOR` operators in the `CONNECT BY` clauses of hierarchical queries.

The following list provides some valid compound expressions:

- `('CLARK' || 'SMITH')`
- `LENGTH('MOOSE') * 57`
- `SORT(144) + 72`
- `my_fun(TO_CHAR(sysdate, 'DD-MMM-YY'))`

17.1.5.7.4. Case expressions

A case expression allows you to use the `IF ... THEN ... ELSE` logic in SQL statements without calling a stored procedure.

Syntax

```
CASE { simple_case_expression
     | searched_case_expression
     }
     [ ELSE else_expr ]
     END
```

The following code provides the syntax of `simple_case_expression`.

```
expr
  { WHEN comparison_expr THEN return_expr }...
```

The following code provides the syntax for `searched_case_expression`:

```
{ WHEN condition THEN return_expr }...
```

Usage rules

Validation of conditions

For a simple case expression, ApsaraDB for OceanBase uses `expr` as a base and searches `WHEN ... THEN` for the first `comparison_expr` that is equal to expr. Then, ApsaraDB for OceanBase returns the corresponding `return_expr`. If no `WHEN ... THEN` meets the condition and an `ELSE` clause exists, ApsaraDB for OceanBase returns `else_expr`. Otherwise, ApsaraDB for OceanBase returns NULL.

In a searched case expression, ApsaraDB for OceanBase searches from left to right until the `condition` is met. Then, ApsaraDB for OceanBase returns `return_expr`. If no condition is met and an `ELSE` clause exists, the database returns `else_expr`. Otherwise, the database returns NULL.

Calculation of conditions

ApsaraDB for OceanBase uses the short-circuit calculation rule. For a simple case expression, the database calculates a `comparison_expr` value only before the database compares the value with `expr`. The database does not calculate all the `comparison_expr` values before the database compares a comparison_expr value with `expr`. Therefore, if the previous `comparison_expr` is equal to `expr`, ApsaraDB for OceanBase does not calculate the next `comparison_expr` value. For a searched case expression, the database performs serial computing for each condition to determine whether the condition is true. If the previous condition is true, ApsaraDB for OceanBase does not calculate the next condition.

Data types

For a simple case expression, the `expr` and all the `comparison_expr` values must have the same data type, such as `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2`, and `NUMBER`, `BINARY_FLOAT`, or `BINARY_DOUBLE`. Alternatively, they must all have the numeric type. If the data type of all the returned expressions is the numeric type, ApsaraDB for OceanBase selects the data type that has the highest precedence. ApsaraDB for OceanBase explicitly converts the other parameters to this data type and returns this data type.

For a simple case expression and a searched case expression, all the `return_exprs` must have the same data type, such as `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2`, and `NUMBER`, `BINARY_FLOAT`, or `BINARY_DOUBLE`. Alternatively, they must all have the numeric type. If the data type of all the returned expressions is the numeric type, ApsaraDB for OceanBase selects the data type that has the highest precedence. ApsaraDB for OceanBase explicitly converts the other parameters to this data type and returns this data type.

Examples

```
SELECT cust_last_name,
    CASE credit_limit
    WHEN 100 THEN 'Low'
    WHEN 5000 THEN 'High'
    ELSE 'Medium' END AS credit
FROM customer
ORDER BY cust_last_name, credit;
```

The following example shows a searched case expression:

```
SELECT AVG(CASE WHEN e.salary > 2000 THEN e.salary
          ELSE 2000 END) "Average Salary" FROM employee e;
```

17.1.5.7.5. Column expressions

Column expressions are a limited form of `expr`. A column expression is named `column_expression` in the syntax of other expressions in this chapter. Column expressions can be simple expressions, compound expressions, function expressions, or expression lists. Column expressions can contain only the following forms of expressions:

- Columns in the table that is created, changed, or indexed.
- Constants (strings or numbers).
- Deterministic functions, such as SQL built-in functions or user defined functions.

Other forms of expressions except the preceding forms of expressions are invalid column expressions. In addition, column expressions do not support compound expressions and aggregate functions that use the `PRIOR` keywords.

You can use column expressions to achieve the following purposes:

- Create a function-based index.
- Explicitly or implicitly define a virtual column. When you define a virtual column, `column_expression` applies to only the column of the table that has been defined in a previous statement.

The components of a column expression must be deterministic. This indicates that the same input values must return the same output values.

17.1.5.7.6. Datetime expressions

A datetime expression generates a value of the datetime data type.

The following syntax is provided:

```
{TIMESTAMP | DATE} string
```

You can combine TIMESTAMP or DATE and a string literal to generate a value of the TIMESTAMP or DATE type. The string format must be consistent with that of the system variables NLS_TIMESTAMP_FORMAT and NLS_DATE_FORMAT.

You can query the values of the system variables by executing the following SQL statement:

```
select * from v$nls_parameters where parameter like '%FORMAT';
```

For example, if the value of the system variable NLS_TIMESTAMP_FORMAT is YYYY-MM-DD HH24:MI:SS.FF, you can use the following expression to generate a value of the TIMESTAMP type.

```
select timestamp '2020-01-01 10:00:00' from dual;
+-------------------------------+
| TIMESTAMP'2020-01-0110:00:00' |
+-------------------------------+
| 2020-01-01 10:00:00.000000000 |
+-------------------------------+
```

17.1.5.7.7. Function expressions

Function expressions allow you to use a built-in SQL function expression or a user defined function expression.

The following list shows some valid built-in function expressions:

- `LENGTH('BLAKE')`
- `ROUND(1234.567*43)`
- `SYSDATE`

> **Notice**
>
> User-defined function expressions cannot pass the parameters of the object type or XMLType to remote functions or procedures.

A user defined function expression calls:

- A function in a custom package or type or in a standalone user defined function
- A user defined function or an operator

The following list shows some valid user defined function expressions:

- `circle_area(radius)`
- `payroll.tax_rate(empno)`
- `hr.employees.comm_pct@remote(dependents, empno)`
- `DBMS_LOB.getlength(column_name)`
- `my_function(a_column)`

If you use user defined functions as expressions, you can use positional notation, named notation, or mixed notation. For example, the following notations are valid:

```
CALL my_function(arg1 => 3, arg2 => 4) ...
CALL my_function(3, 4) ...
CALL my_function(3, arg2 => 4) ...
```

17.1.5.7.8. Interval expressions

An interval expression generates a value of the `INTERVAL YEAR TO MONTH` or `INTERVAL DAY TO SECOND` data type.

The following syntax is provided:

```
INTERVAL string
   { DAY [ (leading_field_precision) ] TO
     SECOND [ (fractional_second_precision) ]
   | YEAR [ (leading_field_precision) ] TO
     MONTH
   }
```

`leading_field_precision` and `fractional_second_precision` can be integers from 0 to 9. The two parameters specify the precisions of the corresponding element values. If you omit the `leading_field_precision` parameter when you specify the `DAY` and `YEAR` elements, the used default value is 2. The value 2 indicates that the value of the element cannot be an integer that has more than two digits. If you omit the following parameter for the `SECOND` element:

`fractional_second_ precision`, the default value is 6. The value 6 indicates that the element value is accurate to the sixth decimal place. If the return value of a query has more digits than the default precision, ApsaraDB for OceanBase returns an error.

For example, the following statement shows an INTERVAL value of the DAY TO SECOND type.

```
select INTERVAL '999999999 23:59:59.999' day(9) to second from dual;
```

17.1.5.7.9. Scalar subquery expressions

A scalar subquery expression is a subquery that returns a row. This row contains multiple column values. If a subquery returns zero rows, the value of the scalar subquery expression is NULL. If a subquery returns multiple rows, the scalar query expression returns an error. In most expressions, scalar subquery expressions can be used as parameters.

Scalar subqueries are not valid expressions in the following scenarios:

- As default values of columns
- In hash functions
- In the `RETURNING` clauses of DML statements
- In the definition of a function index
- In `CHECK` constraints
- In `GROUP BY` clauses
- In the statements that are irrelevant to queries, such as `CREATE PROFILE`

17.1.5.7.10. Expression lists

An expression list is a group of other expressions.

Expression lists can appear in comparison and membership conditions and in the `GROUP BY` clauses of queries and subqueries. Expression lists in comparison and membership conditions are called row value constructors or row constructors.

Comparison and membership conditions appear in the `WHERE` clauses. Comparison and membership conditions can contain one expression or multiple expressions that are separated by commas (,), or one or more groups of expressions. Each group of expressions contains one expression or multiple expressions that are separated by commas (,). In the following examples (multiple groups of expressions):

- Each group is enclosed in parentheses.
- Each group must contain the same number of expressions.
- The number of expressions in each group must match the number of expressions that precede the operator in the comparison condition. Alternatively, the number of expressions in each group must match the number of expressions that precede the `IN` keyword in the membership condition.

A comma-separated list of expressions can contain a maximum of 1,000 expressions. A comma-separated list of groups of expressions can contain expression groups whose quantity is not limited. However, each expression group can contain a maximum of 1,000 expressions.

The following examples show some valid expression lists:

```
(10, 20, 40)
('SCOTT', 'BLAKE', 'TAYLOR')
( ('Guy', 'Himuro', 'GHIMURO'),('Karen', 'Colmenares', 'KCOLMENA') )
```

In the third example, the number of expressions in each group must be the same as the number of expressions in the first part of the condition of the SQL statement. The following example shows the corresponding statement:

```
SELECT * FROM employees
    WHERE (first_name, last_name, email) IN
        (('Guy', 'Himuro', 'GHIMURO'),('Karen', 'Colmenares', 'KCOLMENA'));
```

In a simple `GROUP BY` clause, you can use the uppercase or lowercase expression list.

```
SELECT department_id, MIN(salary) min, MAX(salary) max FROM employees
    GROUP BY department_id, salary
    ORDER BY department_id, min, max;

SELECT department_id, MIN(salary) min, MAX(salary) max FROM employees
    GROUP BY (department_id, salary)
    ORDER BY department_id, min, max;
```

In the `ROLLUP` , `CUBE` , and `GROUPING SETS` clauses of a `GROUP BY` clause, you can combine individual expressions and expression groups in the same expression list. The following example shows some valid expression group lists in the SQL statement:

```
SELECT prod_category, prod_subcategory, country_id, cust_city, count(*)
    FROM products, sales, customers
    WHERE sales.prod_id = products.prod_id
        AND sales.cust_id=customers.cust_id
        AND sales.time_id = '01-oct-00'
        AND customers.cust_year_of_birth BETWEEN 1960 and 1970
    GROUP BY GROUPING SETS (
        (prod_category, prod_subcategory, country_id, cust_city), (prod_category, prod_subcategory, c
ountry_id),        (prod_category, prod_subcategory),
        country_id
        )
    ORDER BY prod_category, prod_subcategory, country_id, cust_city;
```

17.1.5.8. Conditions

17.1.5.8.1. Overview of SQL conditions

Conditions are used to determine data values and return `TRUE`, `FALSE`, or `UNKNOWN`. A condition is a combination of one or more components, such as expressions and logical (Boolean) operators. You must use valid syntax for conditions in SQL statements.

> **Notice**
>
> The `NLS_COMP` and `NLS_SORT` parameters jointly affect the sorting and comparison of characters. If the `NLS_COMP` parameter in the database is set to `LINGUISTIC`, all the entities that are described in this Developer Guide follow the rules that are specified by the `NLS_SORT` parameter. If the `NLS_COMP` parameter is not set to `LINGUISTIC`, the functions are not affected by `NLS_SORT`. You can directly specify the value of `NLS_SORT`. If you do not specify it, it inherits the value of `NLS_LANGUAGE1`.

You can use conditions in the `WHERE` clauses of these statements:

- `DELETE`
- `SELECT`
- `UPDATE`

You can use conditions in these clauses of the `SELECT` statements:

- `WHERE`
- `START WITH`
- `CONNECT BY`
- `HAVING`

A condition can be called a logical data type, although ApsaraDB for OceanBase does not officially support such a data type.

For example, the simple condition 1 = 1 returns the result of TRUE.

The following more complex condition adds the value of **salary** to the value of **commission_pct** and checks whether the sum is greater than 25,000. The `NVL` function replaces NULL values in **salary** with 0.

```
NVL(salary, 0) + NVL(salary + (salary * commission_pct, 0) > 25000)
```

The logical `AND` condition can combine multiple conditions into a single condition.

```
(1 = 1) AND (5 < 7)
```

The following conditions are valid in SQL statements:

```
name = 'SMITH'
employees.department_id = departments.department_id
hire_date > '01-JAN-08'
job_id IN ('SA_MAN', 'SA_REP')
salary BETWEEN 5000 AND 10000
commission_pct IS NULL AND salary = 2100
```

Condition precedence

Condition precedence indicates the order in which ApsaraDB for OceanBase checks different conditions in the same expression. When an expression that contains multiple conditions is computed, the conditions that have higher precedence are checked first, and then the conditions that have lower precedence. The conditions that have equal precedence are checked from left to right. For example, multiple conditions that are connected by `AND` and `OR` cannot be checked from left to right. The `AND` condition is computed first, and then the `OR` condition.

List of SQL condition precedence

Condition type	Functionality
=, ! =, <, >, <=, >=	Comparison
IS [NOT] NULL, LIKE, [NOT] BETWEEN, [NOT] IN, EXISTS, and IS OF	Comparison
NOT	Exponentiation and logical negation
AND	Conjunction
OR	Disjunction

The levels of precedence are listed from high to low. Conditions that are listed on the same line have the same precedence.

17.1.5.8.2. Comparison conditions

Comparison conditions are used to compare an expression with another expression. The result of the comparison is `TRUE`, `FALSE`, or `UNKNOWN`.

You cannot use comparison conditions to compare data of large object (LOB) data types. However, you can compare `CLOB` data by using PL programs.

When numeric expressions are compared, ApsaraDB for OceanBase uses numeric precedence to determine the order in which the numeric values in the conditions are compared. For example, the numeric values of different types appear on both sides of a comparison operator. In this case, if one of the types is `BINARY_DOUBLE`, `BINARY_DOUBLE` values are compared first. Then, `BINARY_FLOAT` values are compared, and finally `NUMBER` values are compared. ApsaraDB for OceanBase determines whether to compare `NUMBER`, `BINARY_FLOAT`, or `BINARY_DOUBLE` values.

When character expressions are compared, ApsaraDB for OceanBase uses specifications in comparison rules of character data types. The rules specify how the character sets of expressions are unified before they are compared, by binary and linguistic comparison, or by blank-padded and nonpadded comparison semantics.

When comparison conditions are used to perform a linguistic comparison on character values, the character values are first converted to sort keys and then compared. The comparison process is similar to that of the `RAW` data type. The sort keys are the values that are returned by the `NLSSORT` function. If the sort keys that are generated by two expressions have the same prefix, the expressions can be linguistically equal even if they differ in the rest of the value.

If two objects of the non-scalar type are of the same named type and a one-to-one correspondence exists between their elements, the two objects are comparable. When you use user-defined object types (nested tables) in equality or `IN` conditions, you must define the `MAP` method. The elements of the nested tables are comparable.

Simple comparison conditions

A simple comparison condition can compare a single expression with an expression list, or compare a single expression with the results of a subquery.

Simple comparison conditions have the following syntax:

```
expr   {= | ! = | ^= | <> | < | >= |}   ( expression_list | subquery )
```

Assume that a single expression is compared with an expression list in a simple comparison condition. The number and data type of the expressions in the expression list must match those of the expressions to the left of the operator. Assume that a single expression is compared with the results of a subquery. The number and data type of the values that are returned by the subquery must match those of the expressions to the left of the operator.

Group comparison conditions

A group comparison condition can compare a single expression with a member or all the members of expression lists or the results of a subquery. The group comparison condition can also compare multiple expressions with a member or all the members of expression lists or the results of a subquery.

Assume that a single expression or multiple expressions are compared with a member or all the members of expression lists in a group comparison condition. The number and data type of the expressions in each expression list must match those of the expressions to the left of the operator. Assume that a single expression or multiple expressions are compared with a member or all the members of the results of a subquery. The number and data type of the values that are returned by the subquery must match those of the expressions to the left of the operator.

Group comparison conditions have the following two types of syntax:

```
expr  {= | ! = | ^= | <> | < | >= | ANY | SOME | ALL } ({ expression_list | subquery})
```

```
(expr [, expr ]...){ = | ! = | ^= | ANY | SOME | ALL} ({expression_list  [, expression_list ]... |sub
query})
```

17.1.5.8.3. Logical conditions

Logical conditions combine two conditions to generate a single result or invert the result of a single condition.

Logical NOT condition

The logical `NOT` condition indicates "not" and can invert the result of a single condition. If the condition is `FALSE` , `TRUE` is returned. If the condition is `TRUE` , `FALSE` is returned. If the condition is `UNKNOWN` , `UNKNOWN` is returned.

Examples of the logical NOT condition

```
SELECT * FROM employees WHERE NOT (job_id IS NULL) ORDER BY employee_id;
SELECT * FROM employees WHERE NOT (salary BETWEEN 1000 AND 2000) ORDER BY employee_id;
```

Logical AND condition

The logical `AND` condition indicates "and". It connects two conditions. If both conditions are `TRUE` , `TRUE` is returned. If either condition is `FALSE` , `FALSE` is returned. Otherwise, `UNKNOWN` is returned.

Examples of the logical AND condition

```
SELECT * FROM employees WHERE job_id = 'PU_CLERK' AND department_id = 30 ORDER BY employee_id;
```

Logical OR condition

The logical `OR` condition indicates "or". This means that either condition is valid. If either condition is `TRUE` , `TRUE` is returned. If both conditions are `FALSE` , `FALSE` is returned. Otherwise, `UNKNOWN` is returned.

Examples of the logical OR condition

```
SELECT * FROM employees WHERE job_id = 'PU_CLERK' OR department_id = 10 ORDER BY employee_id;
```

17.1.5.8.4. Pattern-matching conditions

A pattern-matching condition compares character data.

LIKE condition

A `LIKE` condition is used for pattern matching. The equality operator (=) indicates that one character value exactly matches another character value. The `LIKE` condition matches a portion of one character value with another character value by searching the first value for the pattern that is specified by the second character value.

`LIKE` calculates strings by using the characters that are defined by the input character set.

Syntax

```
char1 [NOT] LIKE char2 [ ESCAPE esc_char ]
```

In addition to `LIKE`, the special pattern-matching character `_` indicates that one character in the value is exactly matched. `%` indicates that zero or multiple characters in the value are matched. The pattern `%` cannot match `NULL`.

Parameters

Parameter	Description
char1	The character expression, such as a character column. It is called the search value.
char2	The character expression that is generally a literal. It is called the pattern.
esc_char	The character expression that is generally a literal. `ESCAPE` converts the `esc_char` identity as an escape character. When an escape character is located before a pattern-matching character, the pattern-matching character is interpreted as a general character.

Examples

The following statement uses the `LIKE` condition:

```
SELECT last_name FROM employees WHERE last_name LIKE '%A\_B%' ESCAPE '\' ORDER BY last_name;
```

`ESCAPE '\'` interprets the pattern-matching character _ after `\` in `%A\_B%` as a general character.

```
SELECT salary FROM employees WHERE 'SM%' LIKE last_name ORDER BY salary;
```

REGEXP_LIKE condition

`REGEXP_LIKE` is used for regular expression matching. `REGEXP_LIKE` evaluates strings by using the characters that are defined by the input character set.

Syntax

```
REGEXP_LIKE(source_char, pattern [, match_param ])
```

Parameters

Parameter	Description
source_char	The string expression that is used as the search value. The data type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.
pattern	The regular expression. The data type can be `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, or `CLOB`.
source_char	The character expression of the `VARCHAR2` or `CHAR` data type. This expression allows you to change the default matching behavior of the condition.

If the data type of the pattern is different from that of `source_char`, ApsaraDB for OceanBase converts the pattern to the data type of `source_char`.

Examples

Create the **employees** table and insert data into the table. Execute the following statements:

```
CREATE TABLE employees(manager_id INT, first_name varchar(50), last_name varchar(50), hiredate varcha
r(50),SALARY INT);
INSERT INTO employees VALUES(300, 'Steven', 'King',  '2019-09-11',23600);
INSERT INTO employees VALUES(200, 'Steven', 'Markle', '2019-11-05', 23800);
INSERT INTO employees VALUES(100, 'Deven',  'Part',  '2018-10-01',24000);
INSERT INTO employees VALUES(200, 'Carlos', 'Ross',  '2019-06-11',23500);
INSERT INTO employees VALUES(200, 'Teven',  'Bell', '2019-05-25', 23000);
INSERT INTO employees VALUES(200, 'Stephen', 'Stiles',  '2018-06-11',24500);
INSERT INTO employees VALUES(100, 'Ame', 'De Haan', '2018-05-01',11000);
INSERT INTO employees VALUES(100, 'Jon', 'Errazuriz', '2017-07-21', 1400);
COMMIT;
```

Query the first names and the last names of the employees whose names contain **Steven** or **Stephen**. `first_name` starts with **Ste** and ends with **en**. **v** or **ph** is in the middle of the first name. Execute the following statement:

```
SELECT first_name, last_name FROM employees WHERE REGEXP_LIKE (first_name, '^Ste(v|ph)en$')
ORDER BY first_name, last_name;
```

The following query result is returned:

```
+------------+-----------+
| FIRST_NAME | LAST_NAME |
+------------+-----------+
| Stephen    | Stiles    |
| Steven     | King      |
| Steven     | Markle    |
+------------+-----------+
```

17.1.5.8.5. NULL conditions

A NULL condition tests null values. This is the only condition that is used to test for null values.

Syntax

```
expr IS [ NOT ] NULL
```

Examples

Execute the following statement:

```
SELECT last_name FROM employees WHERE commission_pct IS NULL
ORDER BY last_name;
```

17.1.5.8.6. Compound conditions

A composition condition is a combination of other conditions.

Syntax

```
{ (condition) | NOT condition | condition { AND | OR } condition }
```

For more information about the `NOT`, `AND`, and `OR` conditions, see the reference document Logical conditions.

17.1.5.8.7. BETWEEN conditions

A `BETWEEN` condition determines whether the value of an expression falls in the interval that is defined by other two expressions.

Syntax

```
expr1 [ NOT ] BETWEEN expr2 AND expr3

NOT (expr1 BETWEEN expr2 AND expr3)
```

The `expr1`, `expr2`, and `expr3` expressions must be the numeric, character, or datetime expressions. In SQL, `expr1` may be calculated for multiple times. If an expression appears in Procedural Language for SQL (PL/SQL), make sure that `expr1` is calculated only once. If the data types of expressions are not the same, ApsaraDB for OceanBase implicitly converts the expressions to the unified data type. Otherwise, the system returns an error.

Examples

Query the information about employees whose salary ranges from 2,000 to 3,000, and sort the result by using employee IDs.

```
SELECT * FROM employees WHERE salary BETWEEN 2000 AND 3000 ORDER BY employee_id;
```

17.1.5.8.8. EXISTS conditions

An EXISTS condition tests whether a specified row exists in a subquery.

Syntax

```
EXISTS (subquery)
```

If a subquery returns at least one row, your required data exists.

Examples

```
SELECT department_id FROM departments d WHERE EXISTS (SELECT * FROM employees e
WHERE d.department_id = e.department_id) ORDER BY department_id;
```

17.1.5.8.9. IN conditions

An `IN` condition is a membership condition. It tests a value or a member value that is in a subquery list.

Syntax

```
expr [ NOT ] IN  ({ expression_list | subquery })
|
( expr [, expr ]... ) [ NOT ]  IN  ( { expression_list [, expression_list ]... | subquery } )
```

An `IN` condition tests whether an expression is a member of an expression list or a subquery, or whether multiple expressions are members of expression lists or subqueries. The expressions in each expression list must match the expressions to the left of the `IN` operator in terms of quantity and data types.

Examples

IN example: equivalent to `=ANY`. It indicates all the members in the set.

```
SELECT * FROM employees WHERE job_id IN ('PU_CLERK','SH_CLERK') ORDER BY employee_id;

SELECT * FROM employees WHERE salary IN (SELECT salary  FROM employees
WHERE department_id =30) ORDER BY employee_id;
```

NOT IN example: equivalent to `! = ALL`. If a member in the set is `NULL`, the calculation result is `false`.

```
SELECT * FROM employees WHERE salary NOT IN (SELECT salary FROM employees
WHERE department_id = 30) ORDER BY employee_id;

SELECT * FROM employees WHERE job_id NOT IN ('PU_CLERK', 'SH_CLERK')
ORDER BY employee_id;
```

17.1.5.9. Queries and subqueries

17.1.5.9.1. Overview of queries and subqueries

A query is a method that is used to retrieve data from databases. It can be used with conditional clauses, such as `WHERE`, and clauses for sorting (such as `ORDER BY`), to retrieve query results. A subquery is a query that is nested in an upper-level query. An upper-level query is called a parent query or an outer query. The result value of the subquery is passed back as an input to the parent query or the outer query. The parent query uses this value in the computation to determine the final output. SQL supports multiple levels of nested queries. This indicates that you can nest other subqueries in a subquery. In addition, subqueries can appear in various clauses of SQL statements, such as `SELECT`, `FROM`, and `WHERE` statements. The following queries are common in SQL statements:

- Simple queries
- Hierarchical queries
- Sets
- Joins
- Subqueries

Simple queries

A simple query is an operation that retrieves data from one or more columns of data from one or more select lists or views in ApsaraDB for OceanBase. The number of columns and their data type and length depend on the elements in the select lists. Select lists are the expression lists that appear after the `SELECT` keyword and before the `FROM` clause.

Hierarchical queries

A hierarchical query is a query statement that has special features. You can use this query to obtain the hierarchical data that is displayed based on hierarchical relationships. Hierarchical data indicates that the data in a relational table has the hierarchical relationship.

Sets

You can use the following set operators to combine multiple queries: `UNION`, `UNION ALL`, `INTERSECT`, and `MINUS`. All the set operators have equal precedence. If an `SQL` statement contains multiple set operators, ApsaraDB for OceanBase checks them from left to right unless an order is specified in the parentheses. This section describes the following set operators.

Operator	Description
UNION	Returns the union of two result sets. The union does not include duplicates.
UNION ALL	Returns the union of two result sets. The union can include duplicates.
INTERSECT	Returns the intersection of two result sets.

Operator	Description
MINUS	Returns the difference of two result sets.

Joins

A join is a query that combines the rows from two or more tables, views, or materialized views. ApsaraDB for OceanBase performs a join when multiple tables appear in the `FROM` clause of the query. The select list of the query can select columns from all of the specified tables. If the same column name is used in the two tables, you must qualify all the references to these columns in the query process by using table names. This section describes the following joins.

Join type	Representation	Description
Equi join	Equijoins	A join that has a join condition in which an equality operator is contained.
Self join	SELF-JOIN	A join of a table to itself.
Inner join	INNER JOIN	An inner join. An inner join returns the matching rows in the two joined tables.
Left (outer) join	LEFT [OUTER] JOIN	The result includes all the rows from the left table (that appears at the leftmost position of the `JOIN` clause) and excludes the rows that do not match from the right table.
Right (outer) join	RIGHT [OUTER] JOIN	The result includes all the rows from the right table (that appears at the rightmost position of the `JOIN` clause) and excludes the rows that do not match from the left table.
Full (outer) join	FULL [OUTER] JOIN	The result includes all the rows from all the joined tables regardless of whether these rows are matched.
Semi join	SEMI-JOIN	You can obtain a semi join by only unnesting a subquery.
Anti join	ANTI-JOIN	Anti joins can be implemented by only subquery unnesting.

Join type	Representation	Description
Cartesian product	Cartesian Products	If two tables are not joined, the data that is retrieved by querying the two tables is the Cartesian product of the two tables.

Subqueries

A subquery is one or more `SELECT` statements that are nested in a `SELECT` query statement. A subquery can return a row, multiple rows, or no result. A subquery in the `FROM` clause of a `SELECT` statement is called an inline view. You can nest one or more subqueries in a nested view. A subquery in the `WHERE` clause of a `SELECT` statement is called a nested subquery.

17.1.5.9.2. Simple queries

A simple query is an operation that retrieves data from one or more columns of data from one or more tables or views in ApsaraDB for OceanBase. The number of columns and their data type and length depend on the schema. Select lists are the expression lists that appear after the `SELECT` keyword and before the `FROM` clause.

Syntax

```
SELECT column name 1,column name 2,column name 3,... FROM table;
```

Table names, field names, and keywords `SELECT` and `FROM` are not case-sensitive. The end of a query can be followed by a semicolon (;). Multiple SQL statements can be executed at the same time. You can use hints in a `SELECT` statement to pass instructions or hints to the ApsaraDB OceanBase database optimizer. The optimizer selects execution plans for statements by using hints.

Examples

Create an employee table **employee** and insert data into the **employee_id**, **first_name**, **last_name**, **manager_id**, and **salary** columns.

```
CREATE TABLE employee (
employee_id INT,
first_name VARCHAR(50),
last_name VARCHAR(50),
manager_id INT,
salary NUMERIC
);
INSERT INTO employee VALUES(111, 'DEL', 'FA BEN', 1, 1500);
INSERT INTO employee VALUES(112, 'AXEL', 'BELL', 1, 1000);
INSERT INTO employee VALUES(113, 'CRIS',  'RACHAR', 1, 1000);
```

Simple query example

- Query some columns:

```
SELECT first_name, last_name, salary FROM employee;
+------------+-----------+--------+
| FIRST_NAME | LAST_NAME | SALARY |
+------------+-----------+--------+
| DEL        | FA BEN    |   1500 |
| AXEL       | BELL      |   1000 |
| CRIS       | RACHAR    |   1000 |
+------------+-----------+--------+
```

- Query all the columns:

```
SELECT * FROM employee;
+-------------+------------+-----------+------------+--------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | MANAGER_ID | SALARY |
+-------------+------------+-----------+------------+--------+
|         111 | DEL        | FA BEN    |          1 |   1500 |
|         112 | AXEL       | BELL      |          1 |   1000 |
|         113 | CRIS       | RACHAR    |          1 |   1000 |
+-------------+------------+-----------+------------+--------+
```

- Perform a mathematical operation on a column:

```
SELECT salary+100 FROM employee;
+-------------+
| SALARY+100 |
+-------------+
|       1600 |
|       1100 |
|       1100 |
+-------------+
```

- Create an alias for a column:

```
SELECT salary*12 Annual salary FROM employee;
+---------+
| Annual salary   |
+---------+
|  18000 |
|  12000 |
|  12000 |
+---------+
```

- Concatenate strings:

```
SELECT first_name || '-' || last_name  AS Full name FROM employee;
+-------------+
| Full name        |
+-------------+
| DEL-FA BEN  |
| AXEL-BELL   |
| CRIS-RACHAR |
+-------------+
```

- Remove duplicates from data

```
SELECT DISTINCT MANAGER_ID FROM employee;
+------------+
| MANAGER_ID |
+------------+
|          1 |
+------------+
```

- `CASE WHEN` statement

```
SELECT salary, CASE WHEN salary >= 1000 then 'High salary >= 800 THEN 'General'
ELSE 'Keeping working hard' END AS typeFROM employee;
+--------+--------+
| SALARY | TYPE   |
+--------+--------+
|   1500 | High salary   |
|   1000 | High salary   |
|   1000 | High salary   |
+--------+--------+
```

17.1.5.9.3. Hierarchical queries

A hierarchical query is a query statement that has special features. You can use this query to display the hierarchical data based on the hierarchical relationship. Hierarchical data indicates that the data in a relational table has the hierarchical relationship. Hierarchical relationships are common in real life. For example, the following hierarchical relationships are common:

- Relationships between team leaders and team members in the organizational structure
- Relationships between superior and subordinate departments in an enterprise
- Relationships between page redirects in web pages

Syntax

```
SELECT [level], column, expr... FROM table [WHERE condition] [ START WITH start_expression ]
CONNECT BY [NOCYCLE] { PRIOR child_expr = parent_expr | parent_expr = PRIOR child_expr }
[ ORDER SIBLINGS BY …] [ GROUP BY … ] [ HAVING … ] [ ORDER BY … ]
```

Parameters

Parameter	Description
LEVEL	The node level. This parameter is a pseudocolumn and specifies the level. The level is counted from the beginning of a query and numbered as 1. The level of the second node in the query is 2, the level for the third node in the query is 3, and so on.
CONNECT_BY_ISLEAF	Specifies whether the current data row is a leaf node in the hierarchical relationship. This parameter is a pseudocolumn. The value 0 indicates that the row is not a leaf node. The value 1 indicates that the row is a leaf node.

Parameter	Description
CONNECT_BY_ISCYCLE	Specifies whether the current data row is in the loop. This parameter is a pseudocolumn. The value 0 indicates that the current data row is not in the loop. The value 1 indicates that the current data row is in the loop.
CONNECT_BY_ROOT operator	CONNECT_BY_ROOT is a unary operator and indicates that the columns in parameters come from the root node of a hierarchical query. This operator has the same precedence as the unary plus (+) and unary minus (-) operators.
condition	The condition.
CONNECT BY	Specifies how to determine the parent-child relationship. An equivalent expression is usually used. Other expressions are also supported.
START WITH	Specifies the root row in the hierarchical query.
PRIOR operator	`PRIOR` is a unary operator and indicates that the columns in parameters come from the parent row. This operator has the same precedence as the unary plus (+) and unary minus (-) operators.
NOCYCLE	When you specify this keyword, the result can still be returned even if the result contains a loop. You can use the `CONNECT_BY_ISCYCLE` virtual column to specify where the loop occurs. When you do not specify this keyword, the client receives an error if a loop occurs.
ORDER SIBLINGS BY	Specifies the order in which rows of the same level are sorted.

Execution process

The key to using and implementing hierarchical queries is to understand their execution processes. The following execution process for a hierarchical query is provided:

1. Perform the `SCAN` or `JOIN` operation that is after `FROM`.
2. Generate the hierarchical relationship results by using the content of `START WITH` and `CONNECT BY`.
3. Execute the remaining clauses, such as `WHERE`, `GROUP`, and `ORDER BY`, based on the regular query execution process. The following process explains how to generate hierarchical relationships in Step 2.
4. Obtain the root rows by using the expression in `START WITH`.
5. Select the child rows of each root row by using the expression in `CONNECT BY`.

6. Use the child rows generated in Step 2 as the new root rows and further generate child rows. This process is repeated until no new rows are generated.

Examples

The following example shows how to use hierarchical queries. Insert data into the **emp_id**, **position**, and **mgr_id** columns of the **emp** table. Execute the following statements:

```
CREATE TABLE emp(emp_id INT,position VARCHAR(50),mgr_id INT);
INSERT INTO emp VALUES (1,'Global Manager',NULL);
INSERT INTO emp VALUES (2,'Europe Regional Manager',1);
INSERT INTO emp VALUES (3,'Asia Pacific Regional Manager',1);
INSERT INTO emp VALUES (4,'Americas Regional Manager',1);
INSERT INTO emp VALUES (5,'Italy Regional Manager',2);
INSERT INTO emp VALUES (6,'France Regional Manager',2);
INSERT INTO emp VALUES (7,'China Regional Manager',3);
INSERT INTO emp VALUES (8,'Korea Regional Manager',3);
INSERT INTO emp VALUES (9,'Japan Regional Manager',3);
INSERT INTO emp VALUES (10,'US Regional Manager',4);
INSERT INTO emp VALUES (11,'Canada Regional Manager',4);
INSERT INTO emp VALUES (12,'Beijing Regional Manager',7);
```

The preceding content shows that data in the **position** column has a clear hierarchical relationship.

To display the results by using the hierarchy, execute the following statement:

```
SELECT emp_id, mgr_id, position, level FROM emp
START WITH mgr_id IS NULL CONNECT BY PRIOR emp_id = mgr_id;
```

The following query result is returned:

```
+--------+--------+-------------------+-------+
| EMP_ID | MGR_ID | POSITION          | LEVEL |
+--------+--------+-------------------+-------+
|      1 |    NULL| Global Manager           |     1 |
|      2 |      1 | Europe Regional Manager         |     2 |
|      5 |      2 | Italy Regional Manager       |     3 |
|      6 |      2 | France Regional Manager         |     3 |
|      3 |      1 | Asia Pacific Regional Manager         |     2 |
|      7 |      3 | China Regional Manager         |     3 |
|     12 |      7 | Beijing Regional Manager         |     4 |
|      8 |      3 | Korea Regional Manager         |     3 |
|      9 |      3 | Japan Regional Manager         |     3 |
|      4 |      1 | Americas Regional Manager         |     2 |
|     10 |      4 | US Regional Manager         |     3 |
|     11 |      4 | Canada Regional Manager       |     3 |
+--------+--------+-------------------+-------+
```

To query the hierarchy of only the Asia Pacific region, execute the following statement:

```
SELECT emp_id, mgr_id, position, level FROM emp START WITH position = 'Asia Pacific Regional Manager'
CONNECT BY PRIOR emp_id = mgr_id;
```

The following query result is returned:

```
+--------+--------+----------------+-------+
| EMP_ID | MGR_ID | POSITION       | LEVEL |
+--------+--------+----------------+-------+
|      3 |      1 | Asia Pacific Regional Manager      |     1 |
|      7 |      3 | China Regional Manager      |     2 |
|     12 |      7 | Beijing Regional Manager      |     3 |
|      8 |      3 | Korea Regional Manager      |     2 |
|      9 |      3 | Japan Regional Manager      |     2 |
+--------+--------+----------------+-------+
```

17.1.5.9.4. Sets

You can use the following set operators to combine multiple queries: `UNION` , `UNION ALL` , `INTERSECT` , and `MINUS` . All the set operators have equal precedence. If an `SQL` statement contains multiple set operators, ApsaraDB for OceanBase checks the operators in the left-to-right order unless an order is specified in the parentheses.

Rules and limits on set operators

You can specify the attribute of `UNION` as `ALL` and `DISTINCT` or `UNIQUE` . ALL indicates that duplicate values are allowed in a set. DISTINCT or UNIQUE indicates that duplicate values are not allowed in a set. However, you cannot specify the `ALL` attribute for other types of set operations. They have only the `DISTINCT` attribute. The default attribute of all set operations is `DISTINCT` . In ApsaraDB for OceanBase, you can specify `ORDER BY` and `LIMIT` clauses in set operations. However, other clauses are not allowed. Only `MINUS` is supported and `EXCEPT` is not supported, although both of them have the same semantics. The number of columns in each query result in a set operation must be the same as that of the corresponding expressions. The corresponding data types must be compatible, such as numeric or character.

Rules of set operators

If component queries query character data, the return values are of the following data types:

- If two values of the `VARCHAR2` type in equal length are queried, the return value is of the `CHAR` type in the same length. If the values of the `CHAR` type in different length are queried, the return value is of the `VARCHAR2` type. Its length is same as that of a larger `CHAR` value.
- If one or two of the queries query the values of the `VARCHAR2` data type, the return value is of the `VARCHAR2` type.

If component queries query numeric data, the data types of the return values are determined by numeric precedence.

- If the values of the `BINARY_DOUBLE` type are queried, the return value is of the `BINARY_DOUBLE` type.
- If the queries select values of the `BINARY_FLOAT` type, the return value is of the `BINARY_FLOAT` type.
- If all the queries select values of the `NUMBER` type, the return value is of the `NUMBER` type.

In queries that use set operators, ApsaraDB for OceanBase does not perform implicit conversion across data type groups. If the corresponding expressions of component queries are resolved into character data and numeric data at the same time, ApsaraDB for OceanBase returns an error.

Limits on set operators

Set operators are subject to the following limits:

- Set operators are invalid on columns of `BLOB` and `CLOB`.
- If the `SELECT` list before the set operator contains an expression, you must provide a column alias for the expression. Then, you can reference the column alias in the `ORDER BY` clause.
- You cannot use set operators to specify `UPDATE` statements.
- You cannot specify `ORDER BY` statements in the subqueries of these operators.
- You cannot use these operators in the `SELECT` statements that contain table collection expressions.

UNION and UNION ALL operators

The `UNION` operator combines the result sets of two or more `SELECT` statements. The `SELECT` statements inside `UNION` must have the same number of columns. The data types of the columns must be similar. In addition, the order of columns in each `SELECT` statement must be the same. By default, the `UNION` operator selects different values. To allow duplicate values, use `UNION ALL`.

Syntax

```
{ (< SQL- Statement 1>) }
UNION [ALL]
{ (< SQL- Statement 2>) }
```

INTERSECT operator

The INTERSECT operator returns the intersection of two result sets, that is, all the distinct values that are returned by both queries.

Syntax

```
{ (< SQL- Statement 1>) }
INTERSECT
{ (< SQL- Statement 2>) }
```

Limits

- The number and order of the columns in all the queries must be same.
- The column data types in the two query result sets that are compared can be different but must be compatible.
- The two query result sets that are compared cannot contain columns of incomparable data types, such as XML, TEXT, NTEXT, IMAGE, or non-binary CLR user-defined types.

- The column names of the returned result set are the same as those returned by the query to the left of the operand. The column names or aliases in `ORDER BY` clauses must reference the column names that are returned by the left-side query.
- The INTERSECT operator cannot be used together with `COMPUTE` and `COMPUTE BY` clauses.
- When rows are compared to determine distinct values, two NULL values are considered equal.

MINUS operator

The `MINUS` operator returns the difference between two result sets, that is, all the distinct values that are returned by the left-side query but not retrieved by the right-side query.

Syntax

```
{ (< SQL- Statement 1>) }
MINUS
{ (< SQL- Statement 2>) }
```

Execution sequence

The following execution sequence is applied when the MINUS operator is used together with other operators in expressions:

1. Expressions in parentheses

2. The `INTERSECT` operand

3. `MINUS` and `UNION` that are evaluated from left to right based on their positions in the expressions

If `MINUS` or `INTERSECT` is used to compare more than two query result sets, the data types are converted based on one comparison between two queries. The preceding rules for evaluating expressions are followed.

Examples

The following statements create tables **table_a** and **table_b** and insert data into the tables:

```
CREATE TABLE table_a(PK INT, name VARCHAR(25));
INSERT INTO table_a VALUES(1,'Fox');
INSERT INTO table_a VALUES(2,'Police');
INSERT INTO table_a VALUES(3,'Taxi');
INSERT INTO table_a VALUES(4,'Lincoln');
INSERT INTO table_a VALUES(5,'New York');
INSERT INTO table_a VALUES(6,'Washington');
INSERT INTO table_a VALUES(7,'Dell');
INSERT INTO table_a VALUES(10,'Lucent');
CREATE TABLE table_b(PK INT, name VARCHAR(25));
INSERT INTO table_b VALUES(1,'Fox');
INSERT INTO table_b VALUES(2,'Police');
INSERT INTO table_b VALUES(3,'Taxi');
INSERT INTO table_b VALUES(6,'Washington');
INSERT INTO table_b VALUES(7,'Dell');
INSERT INTO table_b VALUES(8,'Microsoft');
INSERT INTO table_b VALUES(9,'Apple');
INSERT INTO table_b VALUES(11,'Scotland');
```

Example of UNION:

```
SELECT PK, name FROM table_a
UNION
SELECT PK, name FROM table_b;
```

The following query result is returned:

```
+------+-----------+
| PK   | NAME      |
+------+-----------+
|    1 | Fox    |
|    2 | Police      |
|    3 | Taxi      |
|    4 | Lincoln      |
|    5 | New York      |
|    6 | Washington    |
|    7 | Dell      |
|   10 | Lucent      |
|    8 | Microsoft      |
|    9 | Apple      |
|   11 | Scotland    |
+------+-----------+
```

Example of UNION ALL:

```
SELECT PK, name FROM table_a
UNION ALL
SELECT PK, name FROM table_b;
```

The following query result is returned:

```
+------+-----------+
| PK   | NAME      |
+------+-----------+
|    1 | Fox    |
|    2 | Police      |
|    3 | Taxi      |
|    4 | Lincoln      |
|    5 | New York      |
|    6 | Washington    |
|    7 | Dell      |
|   10 | Lucent      |
|    1 | Fox    |
|    2 | Police      |
|    3 | Taxi      |
|    6 | Washington    |
|    7 | Dell      |
|    8 | Microsoft      |
|    9 | Apple      |
|   11 | Scotland    |
+------+-----------+
```

Example of INTERSECT:

```
SELECT PK, NAME FROM table_a
INTERSECT
SELECT PK, NAME FROM table_b;
```

The following query result is returned:

```
+------+-----------+
| PK   | NAME      |
+------+-----------+
|    1 | Fox       |
|    2 | Police    |
|    3 | Taxi      |
|    6 | Washington|
|    7 | Dell      |
+------+-----------+
```

Example of MINUS:

```
SELECT PK, NAME FROM table_a
MINUS
SELECT PK, NAME FROM table_b;
```

The following query result is returned:

```
+------+--------+
| PK   | NAME   |
+------+--------+
|    4 | Lincoln|
|    5 | New York|
|   10 | Lucent |
+------+--------+
```

17.1.5.9.5. Joins

A join is a query that combines two or more tables, views, or materialized views. If the `FROM` clause of a query contains multiple tables, ApsaraDB for OceanBase performs a join query. The output columns of the query can be selected from a table that is contained in the `FROM` clause. If multiple tables have a same column name, you must use table names to qualify all the references to these columns during the query. The join types in databases generally include `inner join`, `outer join`, `semi-join`, and `anti-join`. Among the preceding join types, you can obtain `semi-join` and `anti-join` by rewriting subqueries. ApsaraDB for OceanBase does not provide syntax for expressing `anti-join` and `semi-join`.

Join conditions

Join conditions are rules to combine multiple tables and exist in FROM clauses or `WHERE` clauses. A join condition is used to compare two columns from different tables. Most joins contain at least one join condition. Join conditions can be divided into equi joins (for example, t1.a = t2.b) and non-equi joins (for example, t1.a < t2.b). Compared with non-equi join conditions, equi join conditions allow databases to use efficient join algorithms, such as `Hash Join` and `Merge-Sort join`.

To perform a join, ApsaraDB for OceanBase retrieves rows from different tables to combine the rows into pairs and matches them based on join conditions. To perform a join on more than two tables, ApsaraDB for OceanBase first joins two of the tables based on the join conditions that compare their columns. Then, ApsaraDB for OceanBase joins the result to another table based on the join conditions that contain the columns of the joined tables and the new table. The optimizer determines the join order in ApsaraDB for OceanBase based on the join conditions, indexes on the base table, and available statistics.

A `WHERE` clause may contain other conditions in addition to join conditions. These conditions that reference only one table can further limit the number of rows returned by the join query.

Equi joins

Equi joins are joins for which join conditions contain equality operators. In an equi join, the rows that meet equality conditions for the specified columns are combined for output.

Self joins

A self join joins a table to itself. The table appears twice in the `FROM` clause and is followed by the table alias that qualify column names in the join condition. If you perform a self join, ApsaraDB for OceanBase combines and returns the rows that meet the join condition.

Cartesian products

If two tables in a join query have no join condition, ApsaraDB for OceanBase returns their Cartesian products. ApsaraDB for OceanBase combines each row of one table with each row of the other table for output. A Cartesian product always generates a number of rows. This is rarely useful. For example, the Cartesian product of two tables that each contain 100 rows is 10,000 rows. Always specify a join condition unless you particularly need a Cartesian product. If a query joins three or more tables and no join condition is specified for a specific pair, the optimizer can choose a join order that prevents from generating an intermediate Cartesian product.

Inner joins

Inner joins are the most basic joins in databases. An inner join combines the columns of two tables, such as Table A and Table B, based on join conditions to generate a new result table. The query compares each row of Table A with each row of Table B and finds combinations that meet the join conditions. If the join conditions are met, the matched rows in Table A and Table B are combined side by side based on columns into one row in the result set. The result set that is generated by the join equals to the result that is generated in the following process: The Cartesian products of the two tables are calculated first. Each row of Table A is combined with each row of Table B. Then, the records that meet the join conditions are returned.

Outer joins

An outer join returns all the rows that meet the join condition, also returns unused rows from one table, and fills in `NULL` at the corresponding positions in the other table. Outer joins can be further divided into left joins, right joins, and full joins. The join type depends on whether the joined table retains rows of the left table, right table, or both tables. In a LEFT [OUTER] JOIN, if the rows in the left table have no matching rows in the right table, the right table is automatically filled with `NULL`. In a RIGHT [OUTER] JOIN, if the rows in the right table have no matching rows in the left table, the left table is automatically filled with `NULL`. In a FULL [OUTER] JOIN, if the rows in the left or right table have no matching rows in the other table, both tables are automatically filled with `NULL`.

Semi-joins

A `LEFT` or `RIGHT ANTI-JOIN` on Table A and Table B returns only the rows in Table A or Table B that have matching rows in Table B or Table A. A `semi-join` can be implemented by only subquery unnesting.

Anti-joins

A `LEFT` or `RIGHT ANTI-JOIN` on Table A and Table B returns all the rows in Table A or Table B that have no matching rows in Table B or Table A. An `anti-join` can be implemented by only subquery unnesting. This is similar to the `semi-join`.

Examples

Create tables **table_a** and **table_b** and insert data. Execute the following statements:

```
CREATE TABLE table_a(PK INT, name VARCHAR(25));
INSERT INTO table_a VALUES(1,'Fox');
INSERT INTO table_a VALUES(2,'Police');
INSERT INTO table_a VALUES(3,'Taxi');
INSERT INTO table_a VALUES(4,'Lincoln');
INSERT INTO table_a VALUES(5,'Arizona');
INSERT INTO table_a VALUES(6,'Washington');
INSERT INTO table_a VALUES(7,'Dell');
INSERT INTO table_a VALUES(10,'Lucent');
CREATE TABLE table_b(PK INT, name VARCHAR(25));
INSERT INTO table_b VALUES(1,'Fox');
INSERT INTO table_b VALUES(2,'Police');
INSERT INTO table_b VALUES(3,'Taxi');
INSERT INTO table_b VALUES(6,'Washington');
INSERT INTO table_b VALUES(7,'Dell');
INSERT INTO table_b VALUES(8,'Microsoft');
INSERT INTO table_b VALUES(9,'Apple');
INSERT INTO table_b VALUES(11,'Scotch whisky');
```

Self join query:

```
SELECT * FROM table_a ta, table_a tb WHERE ta.NAME = tb.NAME;
```

The following query result is returned:

```
+------+-----------------+------+-----------------+
| PK   | NAME            | PK   | NAME            |
+------+-----------------+------+-----------------+
|    1 | Fox             |    1 | Fox             |
|    2 | Police          |    2 | Police          |
|    3 | Taxi            |    3 | Taxi            |
|    4 | Lincoln         |    4 | Lincoln         |
|    5 | Arizona         |    5 | Arizona         |
|    6 | Washington      |    6 | Washington      |
|    7 | Dell            |    7 | Dell            |
|   10 | Lucent          |   10 | Lucent          |
+------+-----------------+------+-----------------+
```

Inner join query:

```
SELECT A.PK AS A_PK, A.name AS A_Value, B.PK AS B_PK, B.name AS B_Value
FROM table_a A INNER JOIN table_b B ON A.PK = B.PK;
```

The following query result is returned:

```
+------+-----------+------+-----------+
| A_PK | A_VALUE   | B_PK | B_VALUE   |
+------+-----------+------+-----------+
|    1 | Fox       |    1 | Fox       |
|    2 | Police    |    2 | Police    |
|    3 | Taxi      |    3 | Taxi      |
|    6 | Washington|    6 | Washington|
|    7 | Dell      |    7 | Dell      |
+------+-----------+------+-----------+
```

Left join query:

```
SELECT  A.PK AS A_PK, A.name AS A_Value, B.PK AS B_PK, B.name AS B_Value
FROM table_a A LEFT JOIN  table_b B ON A.PK = B.PK;
```

The following query result is returned:

```
+------+-----------------+------+-----------+
| A_PK | A_VALUE         | B_PK | B_VALUE   |
+------+-----------------+------+-----------+
|    1 | Fox             |    1 | Fox       |
|    2 | Police          |    2 | Police    |
|    3 | Taxi            |    3 | Taxi      |
|    6 | Washington      |    6 | Washington|
|    7 | Dell            |    7 | Dell      |
|    4 | Lincoln         | NULL | NULL      |
|    5 | Arizona         | NULL | NULL      |
|   10 | Lucent          | NULL | NULL      |
+------+-----------------+------+-----------+
```

Right join query:

```
obclient> SELECT  A.PK AS A_PK, A.name AS A_Value, B.PK AS B_PK, B.name AS B_Value FROM table_a A RIG
HT JOIN   table_b B ON A.PK = B.PK;
```

The following query result is returned:

```
+------+-----------+------+--------------------+
| A_PK | A_VALUE   | B_PK | B_VALUE            |
+------+-----------+------+--------------------+
|    1 | Fox       |    1 | Fox                |
|    2 | Police    |    2 | Police             |
|    3 | Taxi      |    3 | Taxi               |
|    6 | Washington|    6 | Washington         |
|    7 | Dell      |    7 | Dell               |
| NULL | NULL      |    8 | Microsoft          |
| NULL | NULL      |   11 | Scotch whisky      |
| NULL | NULL      |    9 | Apple              |
+------+-----------+------+--------------------+
```

Full join query:

```
obclient> SELECT  A.PK AS A_PK,A.name AS A_Value,B.PK AS B_PK,B.name AS B_Value FROM table_a A FULL
JOIN table_b B ON A.PK = B.PK;
```

The following query result is returned:

```
+------+-----------------+------+--------------------+
| A_PK | A_VALUE         | B_PK | B_VALUE            |
+------+-----------------+------+--------------------+
|    1 | Fox             |    1 | Fox                |
|    2 | Police          |    2 | Police             |
|    3 | Taxi            |    3 | Taxi               |
|    6 | Washington      |    6 | Washington         |
|    7 | Dell            |    7 | Dell               |
| NULL | NULL            |    8 | Microsoft          |
| NULL | NULL            |    9 | Apple              |
| NULL | NULL            |   11 | Scotch whisky      |
|    4 | Lincoln         | NULL | NULL               |
|    5 | Arizona         | NULL | NULL               |
|   10 | Lucent          | NULL | NULL               |
+------+-----------------+------+--------------------+
```

Semi-join: A dependent subquery is unnested and rewritten into a **semi-join**.

```
explain SELECT * FROM table_a t1 WHERE t1.PK IN (SELECT t2.PK FROM table_b t2 WHERE t2.NAME = t1.NAME
);
```

The following query result is returned:

```
+---------------------------------------------+
| Query Plan                                  |
+---------------------------------------------+
======================================
|ID|OPERATOR      |NAME|EST. ROWS|COST|
--------------------------------------
|0 |HASH SEMI JOIN|    |8        |114 |
|1 | TABLE SCAN   |T1  |8        |38  |
|2 | TABLE SCAN   |T2  |8        |38  |
======================================

Outputs & filters:
-------------------------------------
  0 - output([T1.PK], [T1.NAME]), filter(nil),
      equal_conds([T1.PK = T2.PK], [T2.NAME = T1.NAME]), other_conds(nil)
  1 - output([T1.NAME], [T1.PK]), filter(nil),
      access([T1.NAME], [T1.PK]), partitions(p0)
  2 - output([T2.NAME], [T2.PK]), filter(nil),
      access([T2.NAME], [T2.PK]), partitions(p0)
+---------------------------------------------+
```

Anti-join: A dependent subquery is rewritten into an `anti-join`.

```
EXPLAIN SELECT * FROM table_a t1 WHERE t1.PK NOT IN (SELECT t2.PK
FROM table_b t2 WHERE t2.name = t1.name);
```

The following query result is returned:

```
+----------------------------------------------+
| Query Plan                                   |
+----------------------------------------------+
=======================================
|ID|OPERATOR      |NAME|EST. ROWS|COST|
---------------------------------------
|0 |HASH ANTI JOIN|    |0        |112 |
|1 | TABLE SCAN   |T1  |8        |38  |
|2 | TABLE SCAN   |T2  |8        |38  |
=======================================
Outputs & filters:
-------------------------------------
  0 - output([T1.PK], [T1.NAME]), filter(nil),
      equal_conds([T2.NAME = T1.NAME]), other_conds([(T_OP_OR, T1.PK = T2.PK,
      (T_OP_IS, T1.PK, NULL, 0), (T_OP_IS, T2.PK, NULL, 0))])
  1 - output([T1.NAME], [T1.PK]), filter(nil),
      access([T1.NAME], [T1.PK]), partitions(p0)
  2 - output([T2.NAME], [T2.PK]), filter(nil),
      access([T2.NAME], [T2.PK]), partitions(p0)
+----------------------------------------------------------+
```

17.1.5.9.6. Subqueries

A subquery is one or more `SELECT` statements that are nested in a `SELECT` query statement. A subquery can return a row, multiple rows, or no result. A subquery in the `FROM` clause of a `SELECT` statement is called an inline view. A subquery in the `WHERE` clause of a `SELECT` statement is called a nested subquery.

Subqueries are divided into correlated subqueries and uncorrelated subqueries. A correlated subquery is a subquery that depends on the variables of the outer query. This query is usually run for multiple times. An uncorrelated subquery is a subquery that does not depend on the variables of the outer query. This subquery is generally calculated only once. For uncorrelated subqueries and some correlated subqueries, you can rewrite the statements to eliminate subqueries. This achieves the unnesting of the nested subqueries.

Syntax

```
SELECT [ hint ] [ { { DISTINCT | UNIQUE } | ALL } ] select_list
FROM { table_reference | join_clause | ( join_clause ) }
  [ , { table_reference | join_clause | (join_clause) } ]
  [ where_clause ]
  [ hierarchical_query_clause ]
  [ group_by_clause ]
| subquery { UNION [ALL] | INTERSECT | MINUS } subquery [ { UNION [ALL] | INTERSECT | MINUS } subquer
y ]
| ( subquery ) [ order_by_clause ] [ row_limiting_clause ]
```

Parameters

Parameter	Description
select_list	The query list.

Parameter	Description
subquery	The subquery.
hint	The hint.
table_reference	The table to be queried.

If the column in the subquery has the same name as the column in the outer query, you must add a table name or an alias before the duplicate column name in the outer query.

If the upper-level query references the relevant columns in the subquery, the subquery is run. The upper-level query can be a `SELECT`, `UPDATE`, or `DELETE` statement. You can use subqueries in each statement in the following ways:

- Define the rowset that is to be inserted into the table in the `INSERT` or `CREATE TABLE` statement.
- In the `CREATE VIEW` or `CREATE MATERIALIZED VIEW` statement, define the rowset that is to be contained in the view or the materialized view.
- In the `UPDATE` statement, define one or more values to be assigned to the existing rows.
- In the `WHERE` clause, `HAVING` clause, or `START WITH` clause, provide condition values.
- Define the table that contains the query operation.

Unnesting of nested subqueries

Unnesting of nested subqueries is a database optimization strategy. It places some subqueries in the parent query of the outer layer. The essence of this operation is to convert some subqueries into equivalent multi-table join operations. One benefit of this strategy is that it can effectively use the access path, join method, and join order to minimize the number of query layers.

In the following cases, the nested subqueries in the database are unnested:

- Uncorrelated `IN` subqueries.
- The correlated subqueries in `IN` and `EXISTS` do not contain aggregate functions or the `GROUP BY` clauses.

You can use a `UNNEST` hint to control whether to unnest a nested subquery.

Examples

The following statements create the **table_a** table and the **table_b** table and insert data into the tables:

```
CREATE TABLE table_a(PK INT, name VARCHAR(25));
INSERT INTO table_a VALUES(1,'Fox');
INSERT INTO table_a VALUES(2,'Police');
INSERT INTO table_a VALUES(3,'Taxi');
INSERT INTO table_a VALUES(4,'Lincoln');
INSERT INTO table_a VALUES(5,'Arizona');
INSERT INTO table_a VALUES(6,'Washington');
INSERT INTO table_a VALUES(7,'Dell');
INSERT INTO table_a VALUES(10,'Lucent');
CREATE TABLE table_b(PK INT, name VARCHAR(25));
INSERT INTO table_b VALUES(1,'Fox');
INSERT INTO table_b VALUES(2,'Police');
INSERT INTO table_b VALUES(3,'Taxi');
INSERT INTO table_b VALUES(6,'Washington');
INSERT INTO table_b VALUES(7,'Dell');
INSERT INTO table_b VALUES(8,'Microsoft');
INSERT INTO table_b VALUES(9,'Apple');
INSERT INTO table_b VALUES(11,'Scotch whisky');
```

The subquery has no dependency relationship. Execute the following statement:

```
SELECT * FROM TABLE_A T1 WHERE T1.PK IN (SELECT T2.PK FROM TABLE_B T2);
```

The following query result is returned:

```
+------+-----------+
| PK   | NAME      |
+------+-----------+
|    1 | Fox    |
|    2 | Police      |
|    3 | Taxi      |
|    6 | Washington    |
|    7 | Dell      |
+------+-----------+
```

The subquery has the dependency relationship. The outer query variable **T1.PK** is used in the subquery. Execute the following statement:

```
SELECT * FROM TABLE_A T1 WHERE T1.PK IN (SELECT T2.PK FROM TABLE_B T2 WHERE T2.PK = T1.PK);
```

The following query result is returned:

```
+------+-----------+
| PK   | NAME      |
+------+-----------+
|    1 | Fox    |
|    2 | Police      |
|    3 | Taxi      |
|    6 | Washington    |
|    7 | Dell      |
+------+-----------+
```

The subquery that has the dependency relationship is unnested and rewritten as a join. Execute the following statement:

```
EXPLAIN SELECT * FROM TABLE_A T1 WHERE T1.PK IN (SELECT T2.NAME FROM TABLE_B T2 WHERE T2.NAME = T1.NAME);
```

The following query result is returned:

```
+------------------------------------+
| Query Plan                         |
+------------------------------------+
============================================
|ID|OPERATOR            |NAME|EST. ROWS|COST|
--------------------------------------------
|0 |HASH RIGHT SEMI JOIN|    |8        |107 |
|1 | TABLE SCAN         |T2  |8        |38  |
|2 | TABLE SCAN         |T1  |8        |38  |
============================================
Outputs & filters:
-------------------------------------
  0 - output([T1.PK], [T1.NAME]), filter(nil),
      equal_conds([T1.PK = T2.NAME], [T2.NAME = T1.NAME]), other_conds(nil)
  1 - output([T2.NAME]), filter(nil),
      access([T2.NAME]), partitions(p0)
  2 - output([T1.NAME], [T1.PK]), filter(nil),
      access([T1.NAME], [T1.PK]), partitions(p0)
+------------------------------------+
```

17.1.5.10. SQL statements

17.1.5.10.1. DDL

17.1.5.10.1.1. ALTER KEYSTORE

Description

The ALTER KEYSTORE statement modifies keystore attributes, such as enabling or disabling the keystore, changing the keystore password, and generating a secret key for the keystore.

Syntax

```
## Enable or disable KEYSTORE.
ADMINISTER KEY MANAGEMENT SET [keystore_name] OPEN IDENTIFIED BY [password];
ADMINISTER KEY MANAGEMENT SET [keystore_name] CLOSE IDENTIFIED BY [password];

## Set the secret key.
ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY [password]
```

Parameter description

Parameter	Description
keystore_name	Specifies the keystore name.

Parameter	Description
password	Specifies the password of the keystore for access control.

Examples

- Enable the keystore. You can access the encrypted table and set the key only when the keystore is in the open state.

```
ADMINISTER KEY MANAGEMENT SET my_keystore OPEN IDENTIFIED BY abcCBAK123;
```

- Update the master key that is stored in the keystore.

```
ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY abcCBAK123;
```

17.1.5.10.1.2. ALTER OUTLINE

Description

This statement supports only the outlines that you create by using SQL_TEXT. You can execute this statement to add and bind outlines and throttling rules.

Syntax

```
ALTER OUTLINE outline_name ADD stmt [ TO target_stmt ]
```

Parameter description

Parameter	Description
outline_name	The name of the outline to be created.
stmt	The value is generally a DML statement that contains hints and original parameters.

Parameter	Description
TO target_stmt	Assume that you do not specify TO target_stmt and the SQL statement accepted by the database is parameterized. If the parameterized SQL statement is the same as the parameterized text of stmt from which the hint is removed, the SQL statement is bound to the hint in stmt to generate an execution plan. If you need to generate a fixed plan for the statement that contains a hint, you must use TO target_stmt to specify the original SQL statement. **Notice** When target_stmt is used, it is strictly required that stmt exactly matches target_stMT from which the hint is removed.

Examples

- Execute ALTER OUTLINE to add throttling rules.

```
OceanBase (root@oceanbase)> alter outline ol_1 add select /*+max_concurrent(1)*/ * from t1 where c1 =
1 and c2 = ? ;
OceanBase (root@oceanbase)> alter outline ol_1 add select /*+max_concurrent(1)*/ * from t1 where c1 =
? and c2 = 1;
```

- Execute ALTER OUTLINE to add an execution plan.

```
OceanBase (root@oceanbase)> create outline ol_2 on select /*+max_concurrent(1)*/ * from t1,t2 where t
1.c1 = 1;
OceanBase (root@oceanbase)> alter outline ol_2 add select /*+use_nl(t2)*/ * from t1,t2 where t1.c1 =
1;
```

Notes

- You can specify only one execution plan for one outline_name. If you specify an execution plan by executing the create outline statement, you cannot add an execution plan by executing alter outline.
- When you execute alter outline, you can specify only a throttling rule or an execution plan. This is similar to the rule of executing create outline.
- When you execute alter outline, outline_name must match the signature.

17.1.5.10.1.3. ALTER SEQUENCE

Description

This statement modifies an auto-increment column.

Syntax

```
ALTER SEQUENCE sequence_name
  [MINVALUE value | NOMINVALUE]
  [MAXVALUE value | NOMAXVALUE]
  [INCREMENT BY value]
  [CACHE value | NOCACHE]
  [ORDER | NOORDER]
  [CYCLE | NOCYCLE];
```

Parameter description

Parameter	Description
MINVALUE value | NOMINVALUE	Specifies the minimum value of the auto-increment column. Valid values: -10^{27} to (10^{27}-1). If you set this parameter to NOMINVALUE, the minimum value is 1 for the ascending order and the minimum value is -(10^{27}-1) for the descending order. If no parameter is specified, the default value is NOMINVALUE.
MAXVALUE value | NOMAXVALUE	Specifies the maximum value of the auto-increment column. Valid values: (-10^{27}+1) to 10^{27}. If you set this parameter to NOMAXVALUE, the maximum value is (10^{28}-1) for the ascending order and the maximum value is -1 for the descending order. If no parameter is specified, the default value is NOMAXVALUE.
START WITH value	Specifies the start value of the auto-increment column. This value must be less than or equal to MAXVALUE and greater than or equal to MINVALUE. If no parameter is specified, the default value is the minimum value for the ascending order and the default value is the maximum value for the descending order.
INCREMENT BY value	Specifies the auto-increment step for the auto-increment column. The value cannot be 0. If you specify this parameter as a positive number, the auto-increment column is in ascending order. If you specify this parameter as a negative number, the auto-increment column is in descending order. If no parameter is specified, the default value is 1.
CACHE value | NOCACHE	Specifies the number of preassigned auto-increment values in the memory. Default value: 20.

Parameter	Description
ORDER | NOORDER	Specifies whether the values of the auto-increment column are generated in sequence. Default value: NOORDER.
CYCLE | NOCYCLE	Specifies whether the values of the auto-increment column are cyclically generated. Default value: NOCYCLE.

Examples

- Modify the maximum value of the auto-increment column my_sequence and specify that the values of the auto-increment column are cyclically generated.

```
OceanBase(root@oceanbase)>ALTER SEQUENCE my_sequence MAXVALUE 1024 CYCLE;
```

Considerations

- You cannot change the value of START WITH in `ALTER SEQUENCE`. If you need to modify the start position for the next sequence, you can modify `INCREMENT BY` to modify the start position.
- For other considerations of the auto-increment column values, see the CREATE SEQUENCE statement.

17.1.5.10.1.4. ALTER SESSION

Description

This statement modifies the status of a session.

ALTER SESSION has the statements of multiple features. ApsaraDB for OceanBase supports the following three features:

- Switching the database to which the current session connects.
- Modifying the isolation level of the current session.
- Specifying the session variables. You can specify multiple variables in the same statement.

Syntax

```
ALTER SESSION SET CURRENT_SCHEMA = current_schema;
ALTER SESSION SET ISOLATION_LEVEL = [READ UNCOMMITTED|READ COMMITTED|REPEATABLE READ|SERIALIZABLE];
ALTER SESSION SET var1_name = var1_value var2_name = var2_value ... ;
```

Parameter description

Parameter	Description
CURRENT_SCHEMA	Specifies the name of the database to which the connected session switches.

Parameter	Description
ISOLATION_LEVEL	Specifies the isolation level of the session.
var1_name = var1_value var2_name = var2_value ...	Specifies the name and the value of the session variable. When you specify multiple variables, the variables are not separated by commas (,).

Examples

- Change the values of the recyclebin, sql_warnings, and tx_isolation variables of a session.

```
OceanBase(TEST@TEST)>alter session set recyclebin = 'on' sql_warnings = 'on' tx_isolation = 'read-com
mitted';
Query OK, 0 rows affected (0.01 sec)
```

17.1.5.10.1.5. ALTER TABLE

Description

The ALTER TABLE statement changes the schema of an existing table. For example, you can modify a table and table attributes, add columns, modify columns and column attributes, and delete columns.

Syntax

```
alter_table_stmt:
      ALTER TABLE table_name
      alter_table_action_list;
    | RENAME TABLE rename_table_action_list;

alter_table_action_list:
    alter_table_action [, alter_table_action ...]

alter_table_action:
      ADD [COLUMN] {column_definition | (column_definition_list)}
    | MODIFY [COLUMN] column_definition
    | DROP [COLUMN] column_name
    | ADD [CONSTRAINT [constraint_name]] UNIQUE {INDEX | KEY} [index_name] index_desc
        | ADD [CONSTRAINT [constraint_name]] FOREIGN KEY (column_name_list) references_clause
    | ADD [CONSTRAINT [constraint_name]] CHECK (expr)
    | ADD {INDEX | KEY} [index_name] index_desc
    | ADD FULLTEXT [INDEX | KEY] [index_name] fulltext_index_desc
    | ALTER INDEX index_name [VISIBLE | INVISIBLE]
    | DROP {INDEX | KEY} index_name
    | ADD PARTITION (range_partition_list)
    | DROP PARTITION (partition_name_list)
    | REORGANIZE PARTITION name_list INTO partition_range_or_list
    | TRUNCATE PARTITION name_list
    | [SET] table_option_list
    | RENAME [TO] table_name
    | DROP TABLEGROUP
    | DROP CONSTRAINT constraint_name

rename_table_action_list:
    rename_table_action [, rename_table_action ...]
```

```
rename_table_action:
    table_name TO table_name

column_definition_list:
    column_definition [, column_definition ...]

column_definition:
    column_name data_type
    [DEFAULT const_value] [AUTO_INCREMENT]
    [NULL | NOT NULL] [[PRIMARY] KEY] [UNIQUE [KEY]] comment

index_desc:
   (column_desc_list) [index_type] [index_option_list]

fulltext_index_desc:
   (column_desc_list) CTXCAT(column_desc_list) [index_option_list]

column_desc_list:
    column_desc [, column_desc ...]

column_desc:
     column_name [(length)] [ASC | DESC]

index_type:
    USING BTREE

index_option_list:
    index_option [ index_option ...]

index_option:
      [GLOBAL | LOCAL]
    | block_size
    | compression
    | STORING(column_name_list)
    | comment

table_option_list:
    table_option [ table_option ...]

table_option:
    | primary_zone
    | replica_num
    | table_tablegroup
    | block_size
    | compression
    | AUTO_INCREMENT [=] INT_VALUE
    | comment
    | DUPLICATE_SCOPE [=] "none|zone|region|cluster"

partition_option:
      PARTITION BY HASH(expression)
      [subpartition_option] PARTITIONS partition_count
    | PARTITION BY KEY([column_name_list])
      [subpartition_option] PARTITIONS partition_count
    | PARTITION BY RANGE {(expression) | COLUMNS (column_name_list)}
      [subpartition_option] (range_partition_list)

subpartition_option:
      SUBPARTITION BY HASH(expression)
```

```
      SUBPARTITION BY HASH(expression)
      SUBPARTITIONS subpartition_count
    | SUBPARTITION BY KEY(column_name_list)
      SUBPARTITIONS subpartition_count
    | SUBPARTITION BY RANGE {(expression) | COLUMNS (column_name_list)}
      (range_subpartition_list)

range_partition_list:
    range_partition [, range_partition ...]

range_partition:
    PARTITION partition_name
    VALUES LESS THAN {(expression_list) | MAXVALUE}

range_subpartition_list:
    range_subpartition [, range_subpartition ...]

range_subpartition:
    SUBPARTITION subpartition_name
    VALUES LESS THAN {(expression_list) | MAXVALUE}

expression_list:
    expression [, expression ...]

column_name_list:
    column_name [, column_name ...]

partition_name_list:
    partition_name [, partition_name ...]

partition_count | subpartition_count:
    INT_VALUE
```

Parameter description

Parameter	Description
ADD [COLUMN]	Adds a column. You cannot add a primary key column.
MODIFY [COLUMN]	Modifies column attributes.
DROP [COLUMN]	Deletes a column. You are not allowed to delete the primary key column or a column that has indexes.
ADD [UNIQUE INDEX]	Adds a unique index.
ADD [INDEX]	Adds a general index.
ALTER [INDEX]	Modifies index attributes.
ADD [PARTITION]	Adds a partition.

Parameter	Description
DROP [PARTITION]	Deletes a partition.
REORGANIZE [PARTITION]	Reorganizes a partition.
TRUNCATE [PARTITION]	Deletes partition data.
RENAME [TO] table_name	Renames a table.
DROP [TABLEGROUP]	Deletes a table group.
DROP [CONSTRAINT]	Deletes a constraint.
SET BLOCK_SIZE	Specifies the block size of the partitioned table.
SET REPLICA_NUM	Specifies the number of replicas for the table. The value indicates the total number of replicas.
SET COMPRESSION	Specifies the compression method of the table.
SET USE_BLOOM_FILTER	Specifies whether to use BloomFilter.
SET COMMENT	Specifies comment information.
SET PROGRESSIVE_MERGE_NUM	Specifies the number of progressive merge rounds. Valid values: 1 to 64.

Examples

- Change the field type of the field d in table t2.

```
ALTER TABLE t2 MODIFY d CHAR(10);
```

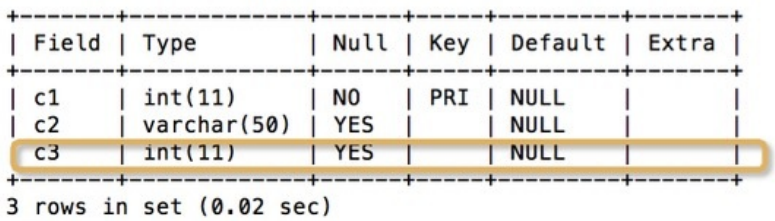
- Add and delete a column.
  - Before you add a column, run the `DESCRIBE test;` command to view table information. The following figure shows the result.

```
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| c1    | int(11)     | NO   | PRI | NULL    |       |
| c2    | varchar(50) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

- Run the following command to add the c3 column:

```
ALTER TABLE test ADD c3 int;
```

- After you add the column, run the `DESCRIBE test;` command to view the table information. The following figure shows the result.

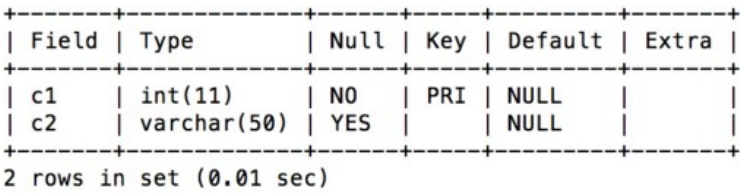


```
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| c1    | int(11)     | NO   | PRI | NULL    |       |
| c2    | varchar(50) | YES  |     | NULL    |       |
| c3    | int(11)     | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.02 sec)
```

- Run the following command to delete the c3 column:

```
ALTER TABLE test DROP c3;
```

- After you delete the column, run the `DESCRIBE test;` command to view the table information. The following figure shows the result.



```
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| c1    | int(11)     | NO   | PRI | NULL    |       |
| c2    | varchar(50) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

- Set the number of replicas for the test table and add the c5 column.

```
ALTER TABLE test SET REPLICA_NUM=2, ADD COLUMN c5 INT;
```

17.1.5.10.1.6. ALTER TABLEGROUP

Description

This statement performs the following operations:

- Add multiple tables to a table group.
- Modify the partitioning rule of a table group.
- Modify the locality and the primary zone of a table group.

Syntax

- Add multiple tables to a table group.

```
ALTER TABLEGROUP tablegroupname ADD [TABLE] tblname [, tblname…]
```

- Modify the partitioning rule of a table group.

```
ALTER TABLEGROUP tablegroupname alter_tg_partition_option
alter_tg_partition_option:
DROP PARTITION '(' name_list ')'
| ADD PARTITION opt_range_partition_list
| modify_tg_partition_info
```

- Modify the locality and the primary zone of a table group.

```
ALTER TABLEGROUP tablegroupname alter_tablegroup_actions
alter_tablegroup_actions:
alter_tablegroup_action
| alter_tablegroup_action, alter_tablegroup_action
alter_tablegroup_action:
SET LOCALITY [=] locality_name
|SET PRIMARY_ZONE [=] primary_zone_name
```

Parameter description

Parameter	Description
tablegroupname	Specifies the table group.
tblname	The table name. If you add multiple tables to the table group, separate the tables with commas (,). If you add multiple tables, duplicate table names are allowed. If the table to be added already exists in the table group that is specified by *tablegroupname* , the system does not report an error.
modify_tg_partition_info	Modifies the partitioning rule of the table group.
LOCALITY locality_name	Specifies the locality of the table group.
PRIMARY_ZONE primary_zone_name	Specifies the primary zone of the table group.

Examples

- Create the table group tgh and two relational tables ttgh and ttgh2. Change the locality of the table group to F@z1.

```
OceanBase(admin@test)> create tablegroup tgh locality='F,R{ALL_SERVER}@z1' partition by hash partitio
ns 10;
Query OK, 0 rows affected (0.09 sec)

OceanBase(admin@test)> create table ttgh(c1 int, c2 int) tablegroup = tgh locality='F,R{ALL_SERVER}@z
1';
Query OK, 0 rows affected (0.55 sec)

OceanBase(admin@test)> create table ttgh2(c1 int, c2 int) tablegroup = tgh locality='F,R{ALL_SERVER}@
z1';
Query OK, 0 rows affected (0.39 sec)

OceanBase(admin@test)> alter tablegroup tgh set locality ='F@z1';
Query OK, 0 rows affected (0.09 sec)

OceanBase(admin@test)> select locality from oceanbase.__all_tablegroup where tablegroup_name ='tgh';
+------------+
| locality   |
+------------+
| FULL{1}@z1 |
+------------+
1 row in set (0.05 sec)

OceanBase(admin@test)> select locality from oceanbase.__all_table where tablegroup_id=(select tablegr
oup_id from oceanbase.__all_tablegroup where tablegroup_name ='tgh');
+------------+
| locality   |
+------------+
| FULL{1}@z1 |
| FULL{1}@z1 |
+------------+
2 rows in set (0.04 sec)
```

17.1.5.10.1.7. ALTER USER

Description

This statement performs the following operations:

- Change the password of an ApsaraDB for OceanBase user.
- Change the encryption method for an ApsaraDB for OceanBase user connection.

> **Note**
>
> To run this command, you must have the UPDATE USER permission.

Syntax

- Change the password of a user.

```
ALTER USER username IDENTIFIED BY password;
```

- Changes the encryption mode for a user connection.

```
ALTER USER user REQUIRE {NONE | SSL | x509 | tls_option_list}

tls_option_list:
  tls_option
  | tls_option_list tls_option

tls_option:
      CIPHER str_value
    | ISSUER str_value
    | SUBJECT str_value
```

Parameter description

Parameter	Description
REQUIRE	Specifies the encryption protocol.

Examples

- Change the password of the sqluser01 user to abc123.

```
ALTER USER sqluser01 IDENTIFIED BY abc123;
```

- Change the encryption protocol for the user connection to SSL.

```
ALTER USER sqluser REQUIRE SSL;
```

17.1.5.10.1.8. CREATE INDEX

Description

This statement creates an index. An index is a structure that is created on a table to sort the values of one or more columns of the database table. The main function of the index is to improve the query speed and reduce the performance cost of the database system.

Syntax

```
CREATE [UNIQUE] INDEX indexname
         ON tblname (index_col_name,...)
      [index_type] [index_options]
index_type:
      USING BTREE

index_options:
      index_option [index_option…]

index_option:
    GLOBAL | LOCAL
    | COMMENT 'string'
    | COMPRESSION [=] {NONE | LZ4_1.0 | LZO_1.0 | SNAPPY_1.0 | ZLIB_1.0}
    | BLOCK_SIZE [=] size
    | STORING(columname_list)
    | VISIBLE | INVISIBLE

index_col_name:
    colname [(length)] [ASC | DESC]

columname_list:
    colname [, colname…]
```

Parameter description

Parameter	Description
indexname	Specifies the name of the index to be created.
tblname	Specifies the name of the table to which the index belongs.
index_col_name	Specifies the column name of the index. Each column name can be followed by ASC that represents the ascending order, and cannot be followed by DESC that represents the descending order. By default, the ascending order is used. In the process of setting up the sorting method of indexes, records are first sorted by using the values of the first column in index_col_name. The records for which the values in the first column are the same are sorted by using the values in the next column. Similar rules apply to the other records.
index_type	The index type. Only USING BTREE is supported. This indicates that B-tree indexes are used.
UNIQUE	Specifies the index as a unique index.
index_option	Specifies the index option. Separate multiple values of index_option with spaces.

Parameter	Description
GLOBAL | LOCAL	Specifies whether the index is a global or local index. Default value: GLOBAL.
COMMENT	Specifies the comment.
COMPRESSION	Specifies the compression algorithm.
BLOCK_SIZE	Specifies the micro-block size.
STORING	Specifies that some columns are stored in the index table for redundant storage. This improves the query performance of systems.

Examples

1. Run the following command to create a table that is named test:

```
CREATE TABLE test (c1 int primary key, c2 VARCHAR(10));
```

2. Run the following command to create an index on the test table:

```
CREATE INDEX test_index ON test (c1, c2 DESC);
```

3. Run the following command to query the index on the test table:

```
SHOW INDEX FROM test;
```

17.1.5.10.1.9. CREATE KEYSTORE

Description

This statement creates a keystore object. This object stores keys. In TDE scenarios, you need to create a keystore object. Each tenant can create only one keystore object.

Syntax

```
ADMINISTER KEY MANAGEMENT CREATE KEYSTORE [KEYSTORE_NAME] IDENTIFIED BY [PASSSWORD] ;
```

Parameter description

Parameter	Description
KEYSTORE_NAME	Specifies the keystore name.

Parameter	Description
PASSWORD	Specifies the password of the keystore for access control.

Examples

Create a keystore object.

```
OceanBase(admin@test)>ADMINISTER KEY MANAGEMENT CREATE KEYSTORE my_keystore IDENTIFIED BY abcCBAK123;
```

17.1.5.10.1.10. CREATE OUTLINE

Description

This statement creates an outline. You can create an outline by using the two methods: SQL_TEXT and SQL_ID. SQL_TEXT is an original statement that contains parameters and is executed by a user.

> **Notice**
>
> To create an outline, you must use the corresponding user for execution.

Syntax

- **Create an outline by using SQL_TEXT.**

```
CREATE [OR REPLACE] OUTLINE outline_name ON stmt [ TO target_stmt ]
```

- **Create an outline by using SQL_ID.**

```
CREATE OUTLINE outline_name ON sql_id USING HINT  hint;
```

Parameter description

Parameter	Description
outline_name	The name of the outline to be created.
OR REPLACE	If the outline to be created already exists after you specify OR REPLACE, the original outline is replaced.
stmt	The value is generally a DML statement that contains hints and original parameters.

Parameter	Description
TO target_stmt	Assume that you do not specify TO target_stmt and the SQL statement accepted by the database is parameterized. If the parameterized SQL statement is the same as the parameterized text of stmt from which the hint is removed, the SQL statement is bound to the hint in stmt to generate an execution plan. If you need to generate a fixed plan for the statement that contains a hint, you must use TO target_stmt to specify the original SQL statement. **Notice** When target_stmt is used, it is strictly required that stmt exactly matches target_stMT from which the hint is removed.
sql_id	If the SQL statement that corresponds to sql_id has a hint, the hint that you specify when you create the outline overwrite all the hints in the original statement.
hint	The format is /*+ xxx */.

Examples

- Create an outline by using SQL_TEXT.

```
CREATE OUTLINE otl_idx_c2
       ON SELECT/*+ index(t1 idx_c2)*/ * FROM t1 WHERE c2 = 1;
```

- Create an outline by using SQL_ID.

```
CREATE OUTLINE otl_idx_c2
ON 'ED570339F2C856BA96008A29EDF04C74'
USING HINT /*+ index(t1 idx_c2)*/ ;
```

Considerations

An outline created by using SQL_TEXT overwrites an outline created by using SQL_ID. SQL_TEXT. The outline created by using SQL_TEXT has the higher precedence.

17.1.5.10.1.11. CREATE SEQUENCE

Description

This statement creates an auto-increment column.

Syntax

```
CREATE SEQUENCE sequence_name
  [MINVALUE value | NOMINVALUE]
  [MAXVALUE value | NOMAXVALUE]
  [START WITH value]
  [INCREMENT BY value]
  [CACHE value | NOCACHE]
  [ORDER | NOORDER]
  [CYCLE | NOCYCLE];
```

Parameter description

Parameter	Description
MINVALUE value | NOMINVALUE	Specifies the minimum value of the auto-increment column. Valid values: -10^{27} to (10^{27}-1). If you set this parameter to NOMINVALUE, the minimum value is 1 for the ascending order and the minimum value is -(10^{27}-1) for the descending order. If no parameter is specified, the default value is NOMINVALUE.
MAXVALUE value | NOMAXVALUE	Specifies the maximum value of the auto-increment column. Valid values: (-10^{27}+1) to 10^{27}. If you set this parameter to NOMAXVALUE, the maximum value is (10^{28}-1) for the ascending order and the maximum value is -1 for the descending order. If no parameter is specified, the default value is NOMAXVALUE.
START WITH value	Specifies the start value of the auto-increment column. This value must be less than or equal to MAXVALUE and greater than or equal to MINVALUE. If no parameter is specified, the default value is the minimum value for the ascending order and the default value is the maximum value for the descending order.
INCREMENT BY value	Specifies the auto-increment step for the auto-increment column. The value cannot be 0. If you specify this parameter as a positive number, the auto-increment column is in ascending order. If you specify this parameter as a negative number, the auto-increment column is in descending order. If no parameter is specified, the default value is 1.
CACHE value | NOCACHE	Specifies the number of preassigned auto-increment values in the memory. Default value: 20.

Parameter	Description
ORDER | NOORDER	Specifies whether the values of the auto-increment column are generated in sequence. Default value: NOORDER.
CYCLE | NOCYCLE	Specifies whether the values of the auto-increment column are cyclically generated. Default value: NOCYCLE.

Examples

- Create an auto-increment column s1.

```
OceanBase(ADMIN@TEST)>create sequence S1 minvalue 1 maxvalue 5 nocycle noorder cache 10240000;
```

Considerations

- If you specify both MINVALUE and MAXVALUE, MINVALUE must be smaller than MAXVALUE.
- The difference between MAXVALUE and MINVALUE must be greater than or equal to INCREMENT BY value.
- The value of the CACHE value must be greater than 1. If the value of CACHE value is 1, the value is equivalent to NOCACHE.
- If you specify CYCLE and INCREMENT BY value is smaller than 0, you must specify MINVALUE.
- If you specify CYCLE, the number of caches cannot exceed one cycle.

17.1.5.10.1.12. CREATE SYNONYM

Description

The CREATE SYNONYM statement creates a synonym.

Syntax

```
CREATE [ OR REPLACE ] [ PUBLIC ]
SYNONYM [ schema. ]synonym
FOR [ schema. ]object;
```

Parameter description

Parameter	Description
OR REPLACE	Specifies that if the name of the synonym to be created already exists, the synonym is recreated based on the new definition.

Parameter	Description
PUBLIC	Specify PUBLIC to create a public synonym that can be used by all users. To use the synonym, users must have the corresponding permission on the basic object. When the reference to an object is resolved, the public synonym is used only if no schema is specified for the object. If you do not specify PUBLIC, the synonym is private and can be accessed by only the current schema. In addition, the synonym name must be unique in the current schema.
[schema.]synonym	schema specifies the schema of the current synonym. If you specify PUBLIC, you do not need to specify a schema for the synonym. synonym specifies the name of the synonym.
[schema.]object	The name of the object that corresponds to the synonym.

Examples

- Create a synonym.

```
OceanBase(TEST@TEST)>create table t1(c1 int);
Query OK, 0 rows affected (0.11 sec)

OceanBase(TEST@TEST)>create synonym s1 for t1;
Query OK, 0 rows affected (0.04 sec)

OceanBase(TEST@TEST)>insert into s1 values(1);
Query OK, 1 row affected (0.04 sec)

OceanBase(TEST@TEST)>select * from s1;
+------+
| C1   |
+------+
|    1 |
+------+
1 row in set (0.02 sec)
```

- Create a public synonym.

```
OceanBase(TEST@TEST)>create public synonym syn_pub for t1;
Query OK, 0 rows affected (0.04 sec)

OceanBase(TEST@TEST)>select * from syn_pub;
+------+
| C1   |
+------+
|    1 |
+------+
1 row in set (0.01 sec)
```

Considerations

To create synonyms, the following permission requirements must be met:

- To create a private synonym in the current schema, you must have the CREATE SYNONYM permission.
- To create a private synonym in a schema that is not the current schema, you must have the CREATE ANY SYNONYM permission.
- To create a public synonym, you must have the CREATE PUBLIC SYNONYM permission.
- The object for which you need to create a synonym does not need to exist. Access permissions are not required.

```
// Create synonym_user.
OceanBase(TEST@TEST)>CREATE USER synonym_user IDENTIFIED BY synonym_user;
Query OK, 0 rows affected (0.04 sec)

OceanBase(TEST@TEST)>grant CREATE on test.* to synonym_user;
Query OK, 0 rows affected (0.05 sec)

OceanBase(TEST@TEST)>grant SELECT on test.* to synonym_user;
Query OK, 0 rows affected (0.02 sec)

// Connect to synonym_user.
OceanBase(SYNONYM_USER@TEST)>create or replace synonym s1 for t1;
ERROR-00600: internal error code, arguments: -5036, Access denied; you need (at least one of) the CRE
ATE SYNONYM privilege(s) for this operation

// Grant the CREATE SYNONYM permission again.
OceanBase(TEST@TEST)>grant CREATE SYNONYM on *.* to synonym_user;
Query OK, 0 rows affected (0.03 sec)

OceanBase(SYNONYM_USER@TEST)>create synonym s1 for t1;
Query OK, 0 rows affected (0.07 sec)
```

17.1.5.10.1.13. CREATE TABLE

Description

The CREATE TABLE statement creates a table in a database.

Syntax

```
CREATE [GLOBAL TEMPORARY] TABLE table_name
      (table_definition_list) [table_option_list] [partition_option] [on_commit_option]
CREATE [GLOBAL TEMPORARY] TABLE table_name
      (table_definition_list) [table_option_list] [partition_option] [AS] select;

table_definition_list:
    table_definition [, table_definition ...]

table_definition:
      column_definition
    | INDEX [index_name] index_desc
    | [CONSTRAINT [constraint_name]] [PRIMARY KEY|UNIQUE]  (column_desc_list) [USING INDEX index_opti
on_list]
    | [CONSTRAINT [constraint_name]] FOREIGN KEY (column_name, column_name ...) references_clause con
stranit_state
    | [CONSTRAINT [constraint_name]] CHECK(expression) constranit_state
```

```
column_definition_list:
    column_definition [, column_definition ...]

column_definition:
    column_name data_type
    [VISIBLE|INVISIBLE]
    {
    [DEFAULT expression]
    [NULL | NOT NULL]
    [CONSTRAINT [constraint_name] [PRIMARY] KEY] [UNIQUE [KEY]]
    [CONSTRAINT [constraint_name] CHECK(expression) constranit_state]
    [CONSTRAINT [constraint_name] references_clause
    |
    [GENERATED ALWAYS] AS (expression) [VIRTUAL]
    [NULL | NOT NULL] [UNIQUE KEY] [[PRIMARY] KEY] [UNIQUE LOWER_KEY] [COMMENT string]
    }

references_clause:
     REFERENCES table_name (column_name, column_name ...) [ON DELETE {CASCADE|SET NULL}]]

constranit_state:
    [RELY|NORELY] [USING INDEX index_option_list] [ENABLE|DISABLE] [VALIDATE|NOVALIDATE]

index_desc:
   (column_desc_list) [index_option_list]

column_desc_list:
    column_desc [, column_desc ...]

column_desc:
     column_name [ASC | DESC][NULL LAST|NULL FIRST]

index_option_list:
    index_option [ index_option ...]

index_option:
      [GLOBAL | LOCAL]
    | block_size
    | compression
    | STORING(column_name_list)
    | comment

table_option_list:
    table_option [ table_option ...]

table_option:
    primary_zone
    | replica_num
    | table_tablegroup
    | block_size
    | compression
    | comment
    | DUPLICATE_SCOPE [=] "none|zone|region|cluster"
    | LOCALITY [=] "locality description"
    | ENABLE ROW MOVEMENT
    | DISABLE ROW MOVEMENT
    | physical_attribute
```

```
physical_attribute_list:
    physical_attribute [physical_attribute]

physical_attribute:
    PCTFREE [=] num
    | PCTUSED num
    | INITRANS num
    | MAXTRANS num
    | STORAGE(storage_option [storage_option] ...)
    | TABLESPACE tablespace


compression:
    NOCOMPRESS
    | COMPRESS { BASIC | FOR OLTP | FOR QUERY [LOW|HIGH]  | FOR ARCHIVE [LOW|HIGH]}

storage_option:
    INITIAL_ num [K|M|G|T|P|E]
    | NEXT num [K|M|G|T|P|E]
    | MINEXTENTS num [K|M|G|T|P|E]
    | MAXEXTENTS num [K|M|G|T|P|E]

partition_option:
      PARTITION BY HASH(column_name_list)
      [subpartition_option] PARTITIONS partition_count [TABLESPACE tablespace] [compression]
    | PARTITION BY RANGE (column_name_list)
      [subpartition_option] (range_partition_list)
    | PARTITION BY LIST (column_name_list)
      [subpartition_option] (list_partition_list)

subpartition_option:
      SUBPARTITION BY HASH (column_name_list) SUBPARTITIONS subpartition_count
    | SUBPARTITION BY RANGE (column_name_list) SUBPARTITION TEMPLATE
      (range_subpartition_list)
    | SUBPARTITION BY LIST (column_name_list) SUBPARTITION TEMPLATE (list_subpartition_list)

range_partition_list:
    range_partition [, range_partition ...]

range_partition:
    PARTITION [partition_name]
    VALUES LESS THAN {(expression_list) | MAXVALUE} [ID = num] [physical_attribute_list] [compression
]

range_subpartition_list:
    range_subpartition [, range_subpartition ...]

range_subpartition:
    SUBPARTITION subpartition_name
    VALUES LESS THAN {(expression_list) | MAXVALUE} [physical_attribute_list]

list_partition_list:
    list_partition [, list_partition] ...

list_partition:
    PARTITION [partition_name] VALUES (DEFAULT|expression_list) [ID num] [physical_attribute_list] [c
ompression]

list_subpartition_list:
```

```
    list_subpartition [, list_subpartition] ...

list_subpartition:
    SUBPARTITION [partition_name] VALUES (DEFAULT|expression_list) [physical_attribute_list]

expression_list:
    expression [, expression ...]

column_name_list:
    column_name [, column_name ...]

partition_name_list:
    partition_name [, partition_name ...]

partition_count | subpartition_count:
    INT_VALUE

on_commit_option:
    ON COMMIT DELETE ROWS
    | ON COMMIT PRESERVE ROWS
```

Parameter description

Parameter	Description
DUPLICATE_SCOPE	Specifies the attributes of the replicated table. Valid values: • none: indicates that the table is a standard table. • zone: indicates that the table is a replicated table. The leader needs to replicate transactions to all the F replicas and R replicas of the current zone. • region: indicates that the table is a replicated table. The leader needs to replicate transactions to all the F replicas and R replicas of the current region. • cluster: indicates that the table is a replicated table. The leader needs to replicate transactions to all the F replicas and R replicas of the cluster. If DUPLICATE_SCOPE is not specified, the default value is none.
BLOCK_SIZE	Specifies the micro-block size of the table.

Parameter	Description
COMPRESSION	Specifies the flat or encoding storage format and the compression method. The following correspondence relationships are available: • nocompress: flat format, none compression • compress [basic]: flat format, lz4_1.0 compression • compress for oltp: flat format, zstd_1.0 compression • query [low|high]: encoding format, lz4_1.0 compression • archive [low|high]: encoding format, zstd_1.0 compression
primary_zone	Specifies the primary zone where the leader replica resides.
replica_num	Specifies the number of replicas.
table_tablegroup	Specifies the tablegroup to which the table belongs.
comment	The comment.
LOCALITY	Describes the distribution of replicas among zones. For example, F@z1,F@z2,F@z3,R@z4 indicates that z1, z2, and z3 are full-featured replicas and z4 is a read-only replica.
physical_attribute	PCTFREE: specifies the percentage of reserved macro block space. Other attributes: Attributes such as STORAGE and TABLESPACE are only for syntax compatibility to facilitate migration and do not take effect.
ENABLE/DISABLE ROW MOVEMENT	Specifies whether to allow data to be moved between partitions.
ON COMMIT DELETE ROWS	Transaction-level temporary tables: The data is deleted on commit.
ON COMMIT PRESERVE ROWS	Session-level temporary tables: The data is deleted when the session ends.

Examples

- Create a database table.

```
CREATE TABLE test (c1 int primary key, c2 VARCHAR(50)) REPLICA_NUM = 3, PRIMARY_ZONE = 'zone1';
```

- Create a replicated table.

```
CREATE TABLE item() locality = 'F,R{all_server}@hz1, F,R{all_server}@hz2,
F,R{all_server}@hz3' DUPLICATE_SCOPE="cluster"
```

- Create a table that has indexes.

```
create table t1 (c1 int primary key, c2 int, c3 int, index i1 (c2));
```

- Create a table that has eight hash partitions.

```
create table t1 (c1 int primary key, c2 int) partition by hash(c1) partitions 8;
```

- Create a table that has range partitions and hash subpartitions.

```
 create table t1 (c1 int, c2 int, c3 int)
     partition by range(c1) subpartition by hash(c2) subpartitions 5
     (partition p0 values less than(0), partition p1 values less than(100));
```

- Enable encoding and zstd compression. Set the percentage of reserved macro block space to 5%.

```
create table t1 (c1 int, c2 int, c3 varchar(64))
  COMPRESS FOR ARCHIVE
  PCTFREE 5;
```

- Create a transaction-level temporary table.

```
create global temporary table t1 (c1 int) on commit delete rows ;
```

- Create a table that has a constraint.

```
create table t1 (c1 int, c2 int, c3 int, CONSTRAINT equal_check CHECK(c2 = c3 * 2) ENABLE VALIDATE);
```

17.1.5.10.1.14. CREATE TABLEGROUP

Description

This statement creates a table group.

> **Note**
>
> Only the tenant administrator can create table groups.

Syntax

```
CREATE TABLEGROUP [IF NOT EXISTS] tablegroupname [opt_tablegroup_option_list] [opt_tg_partition_optio
n]

opt_tablegroup_option_list:
tablegroup_option [tablegroup_option]

tablegroup_option:
LOCALITY [=] locality_name
| PRIMARY_ZONE [=] primary_zone_name

opt_tg_partition_option:
PARTITION BY
 HASH COLUMN_NUM [tg_subpartition_option] PARTITIONS INTNUM
| RANGE COLUMNS COLUMN_NUM [tg_subpartition_option] {PARTITION partition_name VALUES LESS THAN range_
partition_expr, ...}
| LIST COLUMNS COLUMN_NUM [tg_subpartition_option] {PARTITION partition_name VALUES in list_partition
_expr, ...}

tg_subpartition_option:
SUBPARTITION BY
RANGE COLUMN_NUM SUBPARTITION TEMPLATE {SUBPARTITION partition_name VALUES LESS THAN range_partition_
expr, ...}
| HASH COLUMN_NUM [SUBPARTITIONS INTNUM]
| LIST COLUMN_NUM SUBPARTITION TEMPLATE {SUBPARTITION partition_name VALUES in list_partition_expr, .
..}
```

Parameter description

Parameter	Description
tablegroupname	The name of the table group. It must be up to 64 characters in length and can contain only letters, digits, and underscores (_). The name must start with a letter or an underscore (_) and cannot be a keyword that is reserved for ApsaraDB for OceanBase. If the specified table group name is already used and IF NOT EXISTS is not specified, an error occurs.

Parameter	Description
opt_tablegroup_option_list	The partitioning method, locality, and primary zone of the table group must be exactly consistent with those of the tables in the group. You cannot independently change an item of table information in the table group. You can perform only batch operations on the table group. Same locality: The types, quantities, and locations for replicas must be exactly consistent. Same primary zone: The locations and the precedences of leaders must be exactly consistent. Same partitioning method: • The partitioning types must be the same. For example, hash partitioning and range partitioning are used for each table. • If hash partitioning is used, the numbers of referenced columns must be the same and the numbers of partitions must be same. • If range partitioning is used, the number of referenced columns must be the same and the number of partitions must be the same. In addition, the range split points must be the same. • The subpartition requirements must be consistent with the preceding requirements based on the partitioning types.
opt_tg_partition_option	Specifies the partitioning rule for the table group. This rule is the same as the partitioning method that is used by CREATE TABLE. The table group does not have the specific column definition. Therefore, you do not need to write specific columns for HASH, RANGE, and LIST, and you need to specify only the number of columns (COLUMN_NUM).

Examples

- Create a table group that is named myTableGroup1.

```
OceanBase(admin@test)> CREATE TABLEGROUP myTableGroup1;
Query OK, 0 rows affected (0.07 sec)

OceanBase(admin@test)> create table myt1 (c1 int, c2 int ) tablegroup = myTableGroup1;
Query OK, 0 rows affected (0.28 sec)

OceanBase(admin@test)> create table myt2 (c1 int, c2 int ) tablegroup = myTableGroup1;
Query OK, 0 rows affected (0.26 sec)
```

- Create the table group tgh whose partitioning method is hash partitioning. Create the ttgh table whose partitioning method is hash partitioning. The number of partitions for the table group tgh is the same as that for the ttgh table.

```
OceanBase(admin@test)> create tablegroup tgh locality='F,R{ALL_SERVER}@z1' partition by hash partitio
ns 10;
Query OK, 0 rows affected (0.09 sec)

OceanBase(admin@test)> create table ttgh(c1 int, c2 int) locality='F,R{ALL_SERVER}@z1' partition by h
ash(c1) partitions 10;
Query OK, 0 rows affected (0.55 sec)

OceanBase(admin@test)> create table ttgh2(c1 int, c2 int) locality='F,R{ALL_SERVER}@z1' partition by
hash(c2) partitions 10;
Query OK, 0 rows affected (0.39 sec)
```

17.1.5.10.1.15. CREATE TABLESPACE

Description

The CREATE TABLESPACE statement creates a tablespace logical object. The tablespace attributes can be encrypted.

Syntax

```
CREATE TABLESPACE tablespace_name [ENCRYPTION USING 'AES-256|AES-128|AES-192|SM4']
```

Parameter description

Parameter	Description
tablespace_name	Specifies the name of the tablespace object to be created.
ENCRYPTION USING	The keyword for using encryption.
AES-256|AES-128|AES-192|SM4	Represents different encryption algorithms.

Examples

- Create a tablespace object.

```
CREATE TABLESPACE ob_tablespace;

CREATE TABLESPACE ob_tablespace ENCRYPTION USING 'AES-256';
```

17.1.5.10.1.16. CREATE USER

Description

This statement creates a new ApsaraDB for OceanBase user. After a user is created, you can use the user to connect to ApsaraDB for OceanBase.

> **Note**
>
> To use the CREATE USER command, you must have the CREATE USER system permission.

Syntax

```
create_user_stmt:
    CREATE USER user_name [host_name] IDENTIFIED BY password [REQUIRE {NONE | SSL | X509 | tls_option
_list}]
 [PROFILE user_profile] [DEFAULT TABLESPACE table_space]

password:
    STR_VALUE

tls_option_list:
        tls_option
    | tls_option_list, tls_option

tls_option:
          CIPHER STR_VALUE
    | ISSUER STR_VALUE
    | SUBJECT STR_VALUE
```

Parameter description

Parameter	Description
user_name	The username. After a new user is created, a new row for this user is added to the dba_users table. If the same username already exists, the system reports an error.
host_name	The hostname of the user. The value is in the @xxx.xxx.xx.x format.
IDENTIFIED BY	You must use an IDENTIFIED BY clause. You can specify a password for the account.
REQUIRE	Set the encryption protocol used by the user to NONE, SSL, X509, or tls_option_list.
PROFILE user_profile	Specifies a profile that is used by the user.
DEFAULT TABLESPACE table_space	Specifies the default tablespace for the user.

Examples

1. Run the following command to create a user named sqluser for which the password is 123456.

```
CREATE USER sqluser IDENTIFIED BY 123456;
```

2. Run the following command to query the created user.

```
SELECT username FROM dba_users;
```

The following execution result is displayed:

```
OceanBase(TEST@TEST)>CREATE USER sqluser IDENTIFIED BY 123456;
Query OK, 0 rows affected (0.05 sec)
OceanBase(TEST@TEST)>SELECT username FROM dba_users;
+------------+
| USERNAME   |
+------------+
| SYS        |
| LBACSYS    |
| ORAAUDITOR |
| ROOT       |
| TEST       |
| ADMIN      |
| SQLUSER    |
+------------+
7 rows in set (0.01 sec)
```

17.1.5.10.1.17. CREATE VIEW

Description

The CREATE VIEW statement creates a view. If you specify an OR REPLACE clause, you can execute the statement to replace an existing view.

Views actually do not exist in the form of tables in databases and are derived each time the views are used. A view is generated from the result of the SELECT statement that is specified in the CREATE VIEW statement.

Updatable views are supported.

Syntax

```
create_view_stmt:
  CREATE [OR REPLACE] VIEW view_name [(column_name_list)] AS select_stmt;

column_name_list:
    column_name [, column_name ...]
```

Parameter description

Parameter	Description
OR REPLACE	Specifies that if the name of the view to be created already exists, the view is recreated based on the new definition.
view_name	The name of the view.

Parameter	Description
select_stmt	A type of SELECT statements. It defines a view. This statement can select data from base tables or other views.
column_name_list	The view must have unique column names and do not have duplicate column names. This rule is the same as that for base tables. By default, the column names that are retrieved by the SELECT statement are used as the column names for the view. To specify the column names of the view, you can use the optional column_name_list clause and list comma-separated IDs. The number of the column names in the column_name_list clause must be equal to the number of columns that are retrieved by the SELECT statement. The columns that are retrieved by the SELECT statement can be simple references to table columns. They can also be expressions that use functions, constant values, and operators.

Examples

- Select columns c1 and c2 from table t to create a view named v.

```
create or replace view v(vc1, vc2) as select c1, c2 from t;
```

17.1.5.10.1.18. DROP INDEX

Description

The DROP INDEX statement deletes an index. Maintenance overheads increase if excessive indexes exist. Therefore, unnecessary indexes need to be deleted.

When you delete an index, you need to wait for a period before the index is completely deleted.

Syntax

```
DROP INDEX [schema.]indexname;
```

Parameter description

Parameter	Description
schema	Specifies the schema name.
indexname	Specifies the index name.

Examples

- Delete the index test_index.

```
DROP INDEX test_index;
```

17.1.5.10.1.19. DROP OUTLINE

Description

This statement deletes an outline from the ApsaraDB for OceanBase database.

Syntax

```
DROP OUTLINE outline_name;
```

Parameter description

Parameter	Description
outline_name	The name of the outline to be deleted.

Examples

- Delete OUTLINE ol_1.

```
DROP OUTLINE ol_1;
```

17.1.5.10.1.20. DROP SEQUENCE

Description

This statement deletes an auto-increment column.

Syntax

```
DROP SEQUENCE sequence_name
```

Parameter description

Parameter	Description
sequence_name	The name of the auto-increment column to be deleted.

Examples

- Delete the auto-increment column S1.

```
OceanBase(TEST@TEST)>drop sequence S1;
Query OK, 0 rows affected (0.16 sec)
```

17.1.5.10.1.21. DROP SYNONYM

Description

This statement deletes a synonym.

Syntax

```
DROP [PUBLIC] SYNONYM [ schema. ]synonym;
```

Parameter description

Parameter	Description
PUBLIC	Specify PUBLIC to delete a public synonym. If you do not specify PUBLIC, a private synonym is deleted.
[schema.]synonym	The schema field specifies the schema to which the current synonym belongs. If PUBLIC is specified, you do not need to specify the schema field for the synonym. The synonym field specifies the name of the synonym.

Examples

- Delete a synonym.

```
OceanBase(TEST@TEST)>drop synonym test.s1;
Query OK, 0 rows affected (0.04 sec)
```

- Delete a public synonym.

```
OceanBase(TEST@TEST)>drop public synonym syn_pub;
Query OK, 0 rows affected (0.03 sec)
```

Considerations

1. When you delete a synonym, you must have the following permissions:
   - If you need to delete a private synonym:
     - The synonym to be deleted must be under the corresponding schema.
     - You must have the DROP ANY SYNONYM permission.
   - When you delete a public synonym, you must have the DROP PUBLIC SYNONYM permission.
2. When you delete a public synonym, you must specify the PUBLIC keyword and do not specify the schema.

17.1.5.10.1.22. DROP TABLE

Description

This statement deletes tables from an ApsaraDB for OceanBase database.

Syntax

```
DROP TABLE table_name [CASCADE CONSTRAINTS] [PURGE]
```

Parameter description

Parameter	Description
table_name	The name of the table to be deleted.
CASCADE CONSTRAINTS	Performs a cascade delete for the constraints that are associated with table_name.
PURGE	Purges the table. The table is not recycled in the recycle bin.

Examples

- Delete the test table.

```
DROP TABLE test;
```

17.1.5.10.1.23. DROP TABLEGROUP

Description

The DROP TABLEGROUP statement deletes a table group.

Syntax

```
DROP TABLEGROUP [IF EXISTS] tablegroupname
```

Parameter description

Parameter	Description
tablegroupname	The name of the table group. If the name of the table group to be deleted does not exist and IF EXISTS is not specified, an error is reported.

Examples

- Delete the table group named myTableGroup1.

```
OceanBase(admin@test)> DROP TABLEGROUP myTableGroup1;
```

17.1.5.10.1.24. DROP TABLESPACE

Description

The DROP TABLESPACE statement deletes a tablespace logical object. The object can be deleted only if no table exists under the tablespace.

Syntax

```
DROP TABLESPACE tablespace_name;
```

Parameter description

Parameter	Description
tablespace_name	Specifies the name of the tablespace to be deleted.

Examples

- Delete the tablespace object ts.

```
OceanBase(root@oceanbase)>drop tablesapce ts;
```

17.1.5.10.1.25. DROP USER

Description

This statement deletes an ApsaraDB for OceanBase user.

> Note
>
> - To run the DROP USER command, you must have the CREATE USER system permission.
> - CASCADE specifies that before a user is deleted, all the objects for the user, such as permissions, databases, and tables, are deleted.

Syntax

```
DROP USER username CASCADE;
```

Parameter description

Parameter	Description
username	The username. Only a single user can be deleted.

Examples

- Run the following command to delete the user that is named sqluser.

```
oceanBase(admin@TEST)>DROP USER sqluser CASCADE;
Query OK, 0 rows affected (0.06 sec)
```

17.1.5.10.1.26. DROP VIEW

Description

This statement deletes one or more views.

> **Note**
> The current user must have the DROP permission on each view.

Syntax

```
drop_view_stmt:
    DROP VIEW view_name [CASCADE | RESTRICT];
```

Parameter description

Parameter	Description
view_name	The name of the view to be deleted.
CASCADE | RESTRICT	CASCADE and RESTRICT are parsed and ignored.

Examples

- Delete the view v1.

```
OceanBase(root@oceanbase)>drop view v1;
```

17.1.5.10.1.27. RENAME

Description

The RENAME statement renames an object.

Syntax

```
RENAME obj_name TO new_obj_name;
```

Parameter description

Parameter	Description
obj_name	The original name of the object.
new_obj_name	The new name of the object.

Examples

1. Create tables t1 and t2.

```
create table t1(c1 int);
```

2. Rename table t1 to t11.

```
rename t1 to t11;
```

Considerations

You can rename tables, views, private synonyms, and sequences.

17.1.5.10.1.28. TRUNCATE TABLE

Description

The TRUNCATE TABLE statement fully clears a specified table but retains the table schema that includes the defined partition information in the table. This statement is logically the same as the DELETE FROM statement that deletes all rows. To execute the TRUNCATE statement, you must have the permissions to delete and create tables.

The TRUNCATE TABLE and DELETE FROM statements have the following differences:

- The TRUNCATE TABLE operation cancels and recreates the table. This is much faster than deleting rows one after one.
- The result of the TRUNCATE TABLE statement shows that the number of affected rows is always 0.

Syntax

```
TRUNCATE [TABLE] table_name;
```

Parameter description

Parameter	Description
table_name	Specifies the table name.

Examples

- Fully clears table tb1.

```
OceanBase(root@oceanbase)>TRUNCATE TABLE tb1;
```

17.1.5.10.2. DML

17.1.5.10.2.1. DELETE

Description

You can execute the DELETE statement to delete rows that meet the specified conditions from one or more tables.

Syntax

```
DELETE [hint_options] [FROM] table_factor
       [WHERE where_condition]
       [{ RETURNING | RETURN } returning_exprs [into_clause]]

table_factor:
    {tbl_name | table_subquery | '(' table_reference ')' }

where_condition:
    expression

returning_exprs:
    projection [, ...]

into_clause:
{ INTO into_var_list | BULK COLLECT INTO into_var_list}

into_var_list:
{ USER_VARIABLE | ref_name } [, ...]
```

Parameters

Parameter	Description
hint_options	The hint.
table_factor	The name of the table from which you want to delete rows. You can specify a base table, an updatable view, or a special subquery.
where_condition	The filter conditions. The system deletes rows from tables that meet the specified conditions.
returning_exprs	Return the projection that is defined before rows are deleted.
into_clause	Insert the projection that is defined before rows are deleted into the specified table.

Notice

A special subquery is similar to a subquery in an updatable view. A special subquery cannot include complex operators, such as GROUP BY, DISTINCT, and WINDOW FUNCTION.

Examples

The following statements define a sample table and inserts data into the table:

```
OceanBase(admin@test)>create table t1(c1 int primary key, c2 int);
Query OK, 0 rows affected (0.16 sec)
OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |    1 |
|  2 |    2 |
|  3 |    3 |
|  4 |    4 |
+----+------+
4 rows in set (0.06 sec)
```

- Delete rows from a table: Delete the rows where the value in the c1 column is 2. The c1 column is the primary key of the t1 table.

```
OceanBase(admin@test)>DELETE FROM t1 WHERE c1 = 2;
Query OK, 1 row affected (0.02 sec)

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |    1 |
|  3 |    3 |
|  4 |    4 |
+----+------+
3 rows in set (0.01 sec)
```

- Delete rows from a table: Use a subquery to delete rows.

```
OceanBase(admin@test)>DELETE FROM (SELECT * FROM t1);
Query OK, 4 rows affected (0.04 sec)

OceanBase(admin@test)>select * from t1;
Empty set (0.01 sec)
```

- Delete rows from a table: Execute a statement that includes the RETURNING clause.

```
OceanBase(admin@test)>DELETE FROM t1 RETURNING c1;
+----+
| C1 |
+----+
|  1 |
|  2 |
|  3 |
|  4 |
+----+
4 rows in set (0.01 sec)

OceanBase(admin@test)>select * from t1;
Empty set (0.01 sec)
```

17.1.5.10.2.2. INSERT

Description

You can execute the INSERT statement to add one or more records to a table.

Syntax

```
INSERT [hint_options] single_table_insert

single_table_insert:
{ INTO insert_table_clause opt_nologging '(' column_list ')' values_clause [{ RETURNING | RETURN } re
turning_exprs [into_clause]]
| INTO insert_table_clause opt_nologging '(' ')' values_clause [{ RETURNING | RETURN } returning_expr
s [into_clause]]
| INTO insert_table_clause opt_nologging values_clause [{ RETURNING | RETURN } returning_exprs [into_
clause]]
}

opt_nologging: { NOLOGGING | /*EMPTY*/ }

returning_exprs:
    projection [, ...]

into_clause:
{ INTO into_var_list | BULK COLLECT INTO into_var_list}

into_var_list:
{ USER_VARIABLE | ref_name } [, ...]
```

Parameters

Parameter	Description
hint_options	The hint.
insert_table_clause	The table to insert. You can specify a base table, an updatable view, or a special subquery.
opt_nologging	Minimize the number of logs generated when you insert data.
column_list	The columns to insert.
returning_exprs	The projection after you insert data.
into_clause	Insert the updated column values to the specified table columns.

> **Notice**
>
> A special subquery is similar to a subquery in an updatable view. Such a subquery cannot include complex operators, such as GROUP BY, DISTINCT, and WINDOW FUNCTION.

Examples

In the following examples, table t1 is used to describe how to insert data into a single table.

```
OceanBase(admin@test)>create table t1(c1 int primary key, c2 int);
Query OK, 0 rows affected (0.16 sec)
OceanBase(admin@test)>select * from t1;
Empty set (0.02 sec)
```

- Insert a row into table t1.

```
OceanBase(admin@test)>insert into t1 values(1,1);
Query OK, 1 row affected (0.01 sec)

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |    1 |
+----+------+
1 row in set (0.04 sec)
```

- Use a subquery to insert data into table t1.

```
OceanBase(admin@test)>insert into (select * from t1) values(1,1);
Query OK, 1 row affected (0.01 sec)

OceanBase(admin@test)>select * from t1;
+----+------+
| C1 | C2   |
+----+------+
|  1 |    1 |
+----+------+
1 row in set (0.01 sec)
```

- Execute a statement that includes the RETURNING clause to insert data into table t1.

```
OceanBase(admin@test)>insert into t1 values(1,1) returning c1;
+----+
| C1 |
+----+
|  1 |
+----+
1 row in set (0.02 sec)

OceanBase(admin@test)>select * from t1;
+----+------+
| C1 | C2   |
+----+------+
|  1 |    1 |
+----+------+
1 row in set (0.01 sec)
```

17.1.5.10.2.3. MERGE

Description

You can execute the MERGE statement to update data for a destination table based on a source table. For example, you can insert, update, or delete rows.

Syntax

```
MERGE [hint_options]
         INTO table_factor [opt_alias]
        USING table_factor [opt_alias]
            ON '(' expr ')'
    [merge_update_clause]
        [merge_insert_clause]

merge_update_clause:
    WHEN MATCHED THEN UPDATE SET update_asgn_list [WHERE expr] [DELETE WHERE expr]

merge_insert_clause:
    WHEN NOT MATCHED THEN INSERT opt_insert_columns VALUES '(' insert_vals ')' [WHERE expr]
```

Parameters

Parameter	Description
hint_options	The hint.
table_factor	The names of source and destination tables.
ON expr	The JOIN conditions of source and destination tables.
update_asgn_list	Assign values for the update.

Parameter	Description
WHERE expr	The conditions required to trigger the update, delete, or insert operation.

Examples

The following examples are based on tables t1 and t2.

```
create table t1 (c1 int, c2 int);
create table t2 (c1 int, c2 int);

insert into t1 values (0, 0);
insert into t1 values (1, null);
insert into t1 values (2, null);
insert into t2 values (1, 1);
insert into t2 values (2, 20);
insert into t2 values (3, 3);
insert into t2 values (4, 40);
```

Update table t1 based on table t2.

1. Assume that a value in column c1 of table t1 is equal to a value in column c1 of table t2.
    i. If a value in column c2 of table t1 is NULL, the system updates the value by using the value in column c2 of table t2.
    ii. After the update, if the value in column c2 of table t1 is greater than or equal to 0, the system deletes the value.
2. Assume that values in column c1 of table t2 do not match those of table t1.
    i. Find the values in column c2 of table t2 that are less than 10, and insert them into table t1.

```
merge into t1 using t2 on (t1.c1 = t2.c1)
when matched then update set c2 = t2.c2 where t1.c2 is null delete where t1.c2 >= 10
when not matched then insert values (t2.c1, t2.c2) where t2.c2 < 10;
Query OK, 3 rows affected (0.02 sec)

select * from t1;
+------+------+
| C1   | C2   |
+------+------+
|    0 |    0 |
|    1 |    1 |
|    3 |    3 |
+------+------+
```

17.1.5.10.2.4. PURGE DATABASE

Description

You can execute the PURGE DATABASE statement to delete a database from the recycle bin.

Syntax

```
PURGE DATABASE object_name;
```

Parameters

Parameter	Description
object_name	The name of the object in the recycle bin. After an object is moved to the recycle bin, the system generates a new name for the object.

Examples

- Delete database __recycle_$_1_1597384386029184.

```
OceanBase(admin@test)> create database db1;
Query OK, 1 row affected (0.03 sec)

OceanBase(admin@test)> drop database db1;
Query OK, 0 rows affected (0.04 sec)

OceanBase(admin@test)> show recyclebin;
+--------------------------------+---------------+----------+----------------------------+
| OBJECT_NAME                    | ORIGINAL_NAME | TYPE     | CREATETIME                 |
+--------------------------------+---------------+----------+----------------------------+
| __recycle_$_1_1597384386029184 | db1           | DATABASE | 2020-08-14 13:53:06.029367 |
+--------------------------------+---------------+----------+----------------------------+
1 row in set (0.01 sec)

OceanBase(admin@test)> purge database __recycle_$_1_1597384386029184;
Query OK, 0 rows affected (0.03 sec)

OceanBase(admin@test)> show recyclebin;
```

17.1.5.10.2.5. PURGE INDEX

Description

You can execute the PURGE INDEX statement to delete an indexed table from the recycle bin.

Syntax

```
PURGE INDEX object_name;
```

Parameters

Parameter	Description
object_name	The name of the object in the recycle bin. After an object is moved to the recycle bin, the system generates a new name for the object.

Examples

- Delete indexed table __recycle_$_1_1597387726700872 from the recycle bin.

```
OceanBase(admin@test)> create table t1(c1 int);
Query OK, 0 rows affected (0.09 sec)

OceanBase(admin@test)> create index idx on t1(c1);
Query OK, 0 rows affected (0.48 sec)

OceanBase(admin@test)> drop table t1;
Query OK, 0 rows affected (0.03 sec)

OceanBase(admin@test)> show recyclebin;
+--------------------------------+----------------------------+-------+----------------------------+
| OBJECT_NAME                    | ORIGINAL_NAME              | TYPE  | CREATETIME                 |
+--------------------------------+----------------------------+-------+----------------------------+
| __recycle_$_1_1597387726700872 | __idx_1101710651081557_idx | INDEX | 2020-08-14 14:48:46.699145 |
| __recycle_$_1_1597387726712976 | t1                         | TABLE | 2020-08-14 14:48:46.712643 |
+--------------------------------+----------------------------+-------+----------------------------+
5 rows in set (0.01 sec)

OceanBase(admin@test)> purge index __recycle_$_1_1597387726700872;
Query OK, 0 rows affected (0.04 sec)
```

17.1.5.10.2.6. PURGE RECYCLEBIN

Description

You can execute the PURGE RECYCLEBIN statement to empty the recycle bin as the root user.

> **Notice**
>
> The PURGE RECYCLEBIN statement deletes all objects from the recycle bin. You can execute this statement as the root user only. Proceed with caution.

Syntax

```
PURGE RECYCLEBIN;
```

Parameters

None

Examples

- Empty the recycle bin as the root user.

```
OceanBase(admin@test)> purge recyclebin;
Query OK, 0 rows affected (0.03 sec)
```

17.1.5.10.2.7. PURGE TABLE

Description

You can execute the PURGE TABLE statement to delete a table from the recycle bin.

Syntax

```
PURGE TABLE object_name;
```

Parameters

Parameter	Description
object_name	The name of the object in the recycle bin. After an object is moved to the recycle bin, the system renames the object.

Examples

- Delete table __recycle_$_1_1099511628776_1099511677778 from the recycle bin.

```
OceanBase(admin@test)> create table test(c1 int);
Query OK, 0 rows affected (0.16 sec)

OceanBase(admin@test)> drop table test;
Query OK, 0 rows affected (0.03 sec)

OceanBase(admin@test)> show recyclebin;
+-------------------------------------------+---------------+-------+----------------------------+
| OBJECT_NAME                               | ORIGINAL_NAME | TYPE  | CREATETIME                 |
+-------------------------------------------+---------------+-------+----------------------------+
| __recycle_$_1_1099511628776_1099511677778 | test          | TABLE | 2017-10-20 17:40:22.304025 |
+-------------------------------------------+---------------+-------+----------------------------+
1 row in set (0.02 sec)

OceanBase(admin@test)> purge table __recycle_$_1_1099511628776_1099511677778;
Query OK, 0 rows affected (0.04 sec)
```

17.1.5.10.2.8. SELECT

The syntax of SELECT statements is complex. This topic describes the syntax of simple SELECT statements, SELECT statements that contain set operators, and SELECT statements that contain WITH clauses.

SIMPLE SELECT

Description

You can execute a simple SELECT statement to query table data.

Syntax

```
simple_select:
SELECT [/*+ hint statement */] [DISTINCT | UNIQUE | ALL]
    select_expr_list FROM from_list [WHERE condition]
  [GROUP BY group_expression_list] [ROLLUP group_expression_list] [HAVING condition]]
  [ORDER BY order_expression_list]
  [FOR UPDATE]

select_expr:
    table_name.*
  | table_alias_name.*
  | expr [[AS] column_alias_name]

from_list:
    table_reference [, table_reference ...]

table_reference:
      simple_table
    | joined_table

simple_table:
    table_factor [partition_option] [[AS] table_alias_name]
    | (select_stmt) [AS] table_alias_name
    | (table_reference_list)

joined_table:
      table_reference [INNER] JOIN simple_table [join_condition]
    | table_reference outer_join_type JOIN simple_table join_condition

partition_option:
    PARTITION (partition_name_list)

partition_name_list:
    partition_name [, partition_name ...]

outer_join_type:
    {LEFT | RIGHT | FULL} [OUTER]

join_condition:
    ON expression

condition:
    expression

group_expression_list:
    group_expression [, group_expression ...]

group_expression:
    expression [ASC | DESC]

order_expression_list:
    order_expression [, order_expression ...]

order_expression:
    expression [ASC | DESC]
```

Parameters

Parameter	Description
DISTINCT UNIQUE ALL	Specifies whether to return distinct table rows. A database table may contain duplicate rows. • If you add DISTINCT to the statement, only distinct rows are returned. • If you add UNIQUE to the statement, only distinct rows are returned. • If you add ALL to the statement, all the matched rows are returned, including the duplicate rows. By default, ALL is used.
select_expr	The expressions or column names to query. To specify multiple expressions or column names, separate them with commas (,). You can use an asterisk (*) to query all the columns.
AS othername	Rename the output fields.
FROM table_references	The tables from which you want to query data. You can query data from multiple tables.
WHERE where_conditions	The filter conditions. The query results contain the rows that match the filter conditions. This clause is optional. where_conditions specifies an expression.
GROUP BY group_by_list	Group data based on specified fields and generate statistics.
ROLLUP group_expression_list	Merge the groups generated by the Group By statement and generate statistics.
HAVING search_confitions	The filter conditions. HAVING clauses are similar to WHERE clauses. The difference between them is that you can use aggregate functions in HAVING clauses, such as SUM and AVG.
ORDER BY order_list order_list : colname [ASC | DESC] [,colname [ASC | DESC]...]	Display the query results in ascending or descending order. ASC indicates the ascending order and DESC indicates the descending order. If you do not specify the order, the default order ASC is used.

Parameter	Description
FOR UPDATE	Place an exclusive lock on each row of the query results. This prevents other transactions from concurrently updating the rows. This also prevents other transactions from concurrently reading the rows for which some transaction isolation levels are specified.
PARTITION(partition_list)	The partition information of the specified tables. Format: partition(p0,p1...)

Examples

The following examples are based on table a.

Table a

id	name	num
1	a	100
2	b	200
3	a	50

- Query the name column from table a.

```
SELECT name FROM a;
```

```
+------+
| name |
+------+
| a    |
| b    |
| a    |
+------+
3 rows in set (0.01 sec)
```

- Return the distinct rows for the name column.

```
SELECT DISTINCT name FROM a;
```

```
+------+
| name |
+------+
| a    |
| b    |
+------+
2 rows in set (0.01 sec)
```

- In table a, query the id, name, and num columns, divide the num column values by 2, and return the modified the num column as a column named avg.

```
SELECT id, name, num/2 AS avg FROM a;
```

```
+------+------+------+
| id   | name | avg  |
+------+------+------+
|    1 | a    |   50 |
|    2 | b    |  100 |
|    3 | a    |   25 |
+------+------+------+
3 rows in set (0.00 sec)
```

- In table a, find the rows in which the value for the name column is 'a'. Then, return the values at the intersections of this row and the id, name, and num columns.

```
SELECT id, name, num FROM a WHERE name = 'a';
```

```
+------+------+------+
| id   | name | num  |
+------+------+------+
|    1 | a    |  100 |
|    3 | a    |   50 |
+------+------+------+
2 rows in set (0.01 sec)
```

- In table a, query the id and name columns, and group the num column values by name. Then, return the sum of the num column values in each group.

```
SELECT id, name, SUM(num) FROM a GROUP BY name;
```

```
+------+------+----------+
| id   | name | SUM(num) |
+------+------+----------+
|    1 | a    |      150 |
|    2 | b    |      200 |
+------+------+----------+
2 rows in set (0.00 sec)
```

- In table a, query the id and name columns, group the num column values by name, and calculate the sum of the num column values in each group. Then, return the sum value that is less than 160.

```
SELECT id, name, SUM(num) as sum FROM a GROUP BY name HAVING SUM(num) < 160;
```

```
+------+------+------+
| id   | name | sum  |
+------+------+------+
|    1 | a    |  150 |
+------+------+------+
1 row in set (0.01 sec)
```

- In table a, query the id, name, and num columns. Then, sort the data by the num column in ascending order and return the result set.

```
SELECT * FROM a ORDER BY num ASC;
```

```
+------+------+------+
| id   | name | num  |
+------+------+------+
|    3 | a    |   50 |
|    1 | a    |  100 |
|    2 | b    |  200 |
+------+------+------+
3 rows in set (0.00 sec)
```

- In table a, query the id, name, and num columns. Then, sort the data by the num column in descending order and return the result set.

```
SELECT * FROM a ORDER BY num DESC;
```

```
+------+------+------+
| id   | name | num  |
+------+------+------+
|    2 | b    |  200 |
|    1 | a    |  100 |
|    3 | a    |   50 |
+------+------+------+
3 rows in set (0.01 sec)
```

SELECT statements that contain set operators

Description

You can execute the SELECT statement that contains the UNION, MINUS, or INTERSECT operator to combine query results.

Syntax

```
select_clause_set:
  simple_select [ UNION | UNION ALL |  | INTERSECT] select_clause_set_right
  [ORDER BY sort_list_columns]

select_clause_set_right:
  simple_select |
  select_caluse_set
```

Parameters

Parameter	Description
UNION ALL	Combine the results of two queries and return all rows.
UNION	Combine the results of two queries and return distinct rows.
MINUS	Return the distinct rows in the left query that are not selected by the right query.

Parameter	Description
INTERSECT	Return the distinct rows that are selected by the left and right queries.

Examples

The following examples are based on tables t1 and t2.

```
create table t1 (c1 int, c2 int);
create table t2 (c1 int, c2 int);
insert into t1 values (1, -1), (2, -2);
insert into t2 values (1, 1), (2, -2), (3, 3);
```

- Query all rows from t1 and t2.

```
SELECT C1, C2 FROM T1 UNION ALL SELECT C1, C2 FROM T2;
+------+------+
| C1   | C2   |
+------+------+
|    1 |   -1 |
|    2 |   -2 |
|    1 |    1 |
|    2 |   -2 |
|    3 |    3 |
+------+------+
```

- Query all distinct rows from t1 and t2.

```
SELECT C1, C2 FROM T1 UNION SELECT C1, C2 FROM T2;
+------+------+
| C1   | C2   |
+------+------+
|    1 |   -1 |
|    2 |   -2 |
|    1 |    1 |
|    3 |    3 |
+------+------+
```

- Query the rows that exist in t1 and t2.

```
SELECT C1, C2 FROM T1 INTERSECT SELECT C1, C2 FROM T2;
+------+------+
| C1   | C2   |
+------+------+
|    2 |   -2 |
+------+------+
```

- Query the rows that exist in t1 but do not exist in t2.

```
SELECT C1, C2 FROM T1 MINUS SELECT C1, C2 FROM T2;
+------+------+
| C1   | C2   |
+------+------+
|    1 |   -1 |
+------+------+
```

SELECT statements that contain WITH clauses

Description

You can execute the SELECT statement that contains a WITH clause to reduce duplicate subqueries. Subqueries in WITH clauses are used as common expressions that can be referenced by each query.

Syntax

```
with_clause_select:
    with_clause simple_select

with_clause:
    WITH table_name [opt_column_alias_name_list] AS ( select_clause )

select_clause:
    simple_select | select_clause_set

opt_column_alias_name_list:
    (column_name_list)

column_name_list:
    column_name | column_name , column_name_list
```

Parameters

None

Examples

- The following examples are based on tables t1 and t2. Execute the following SELECT statement to query t1 and t2:

```
create table t1(c1 int, c2 int, c3 int);
create table t2(c1 int);
insert into t1 values(1,1,1);
insert into t1 values(2,2,2);
insert into t1 values(3,3,3);
insert into t2 values(4);

select * from t1 where c1 > (select count(*) from t2)
                     and c2 > (select count(*) from t2)
                     and c3 > (select count(*) from t2);
+------+------+------+
| C1   | C2   | C3   |
+------+------+------+
|    2 |    2 |    2 |
|    3 |    3 |    3 |
+------+------+------+
```

Extract duplicate subqueries from the preceding SELECT statement, and add the subquery to a WITH clause.

```
with temp(cnt) as (select count(*) from t2)
select t1.* from t1, temp where c1 > temp.cnt and c2 > temp.cnt and c3 > temp.cnt;
+------+------+------+
| C1   | C2   | C3   |
+------+------+------+
|    2 |    2 |    2 |
|    3 |    3 |    3 |
+------+------+------+
```

17.1.5.10.2.9. UPDATE

Description

You can execute the UPDATE statement to change field values in a table.

Syntax

```
UPDATE [hint_options] dml_table_clause
       SET update_asgn_list
       [WHERE where_condition]
       [{ RETURNING | RETURN } returning_exprs [into_clause]]

dml_table_clause:
    dml_table_name opt_table_alias

update_asgn_list:
    column_name = expr [, ...]

where_condition:
    expression

returning_exprs:
    projection [, ...]

into_clause:
{ INTO into_var_list | BULK COLLECT INTO into_var_list}

into_var_list:
{ USER_VARIABLE | ref_name } [, ...]
```

Parameters

Parameter	Description
hint_options	The hint.
dml_table_clause	The name of the table that you want to update. You can specify a base table, an updatable view, or a special subquery.
where_condition	The filter conditions.

Parameter	Description
update_asgn_list	The columns to update.
returning_exprs	The projection after you update data.
into_clause	After you update the table, insert the projection into the specified table columns.

Notice

A special subquery is similar to a subquery in an updatable view. Such a subquery cannot include complex operators, such as GROUP BY, DISTINCT, or WINDOW FUNCTION.

Examples

Create sample tables t1 and t2.

```
OceanBase(admin@test)>create table t1(c1 int primary key, c2 int);
Query OK, 0 rows affected (0.16 sec)
OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |    1 |
|  2 |    2 |
|  3 |    3 |
|  4 |    4 |
+----+------+
4 rows in set (0.06 sec)
```

- Update a single table: In the t1 table, find the row that matches the t1.c1 = 1 condition, and change the value at the intersection of this row and the c2 column to 100.

```
OceanBase(admin@test)>update t1 set t1.c2 = 100 where t1.c1 = 1;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

OceanBase(admin@test)>select * from t1;
+----+------+
| c1 | c2   |
+----+------+
|  1 |  100 |
|  2 |    2 |
|  3 |    3 |
|  4 |    4 |
+----+------+
4 rows in set (0.01 sec)
```

- Update a single table: Use a subquery to query table v. Find the row that matches the v.c1 = 1 condition in table v, and change the value at the intersection of this row and the c2 column to 100.

```
OceanBase(admin@test)>update (select * from t1)v set v.c2 = 100 where v.c1 = 1;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

OceanBase(admin@test)>select * from t1;
+----+------+
| C1 | C2   |
+----+------+
|  1 |  100 |
|  2 |    2 |
|  3 |    3 |
|  4 |    4 |
+----+------+
4 rows in set (0.01 sec)
```

- Update a single table: Execute a statement that includes the RETURNING clause.

```
OceanBase(admin@test)>update t1 set t1.c2 = 100 where t1.c1 = 1 returning c2;
+------+
| C2   |
+------+
|  100 |
+------+
1 row in set (0.02 sec)

OceanBase(admin@test)>select * from t1;
+----+------+
| C1 | C2   |
+----+------+
|  1 |  100 |
|  2 |    2 |
|  3 |    3 |
|  4 |    4 |
+----+------+
4 rows in set (0.01 sec)
```

17.1.5.10.3. DCL

17.1.5.10.3.1. EXPLAIN

Description

This statement explains the execution plan of an SQL statement, such as a SELECT, DELETE, INSERT, REPLACE, or UPDATE statement.

Syntax

```
Retrieve the information about a table or a column:
{EXPLAIN | DESCRIBE | DESC} tbl_name [col_name | wild]

Retrieve the information about an SQL plan:
{EXPLAIN}
[BASIC | OUTLINE | EXTENDED | EXTENDED_NOADDR | PARTITIONS | FORMAT = {TRADITIONAL| JSON}]
{SELECT statement | DELETE statement | INSERT statement | UPDATE statement | MERGE statement}
```

Parameter description

Parameter	Description
tbl_name	Specifies the table name.
col_name	Specifies the column name of the table.
BASIC	Specifies the basic information about the output plan, such as the operator ID, operator name, and referenced table name.
OUTLINE	Specifies that the output plan information includes the outline information.
EXTENDED	Specifies that the EXPLAIN statement generates additional information. The additional information includes the input and output columns for each operator, the partition information about the accessed table, and the current used filter information. If the current operator uses an index, the used index column and the extracted query range appear.
EXTENDED_NOADDR	Displays the additional information in a simple way.
PARTITIONS	Displays the partition-related information.
FORMAT = {TRADITIONAL| JSON}	Specifies the output format of EXPLAIN: • TRADITIONAL: the table output format. • The KEY:VALUE output format. The output appears as JSON strings that contain EXTENDED and PARTITIONS information.

Examples

- Omit explain_type

```
OceanBase(admin@test)>explain select * from t1,t2 where t1.c2=t2.c2 and t2.c1 > 4\G
*************************** 1. row ***************************
Query Plan: ========================================
|ID|OPERATOR   |NAME|EST. ROWS|COST   |
----------------------------------------
|0 |HASH JOIN  |    |9801000  |5933109|
|1 | TABLE SCAN|t2  |10000    |6219   |
|2 | TABLE SCAN|t1  |100000   |68478  |
========================================

Outputs & filters:
-------------------------------------
  0 - output([t1.c1], [t1.c2], [t2.c1], [t2.c2]), filter(nil),
      equal_conds([t1.c2 = t2.c2]), other_conds(nil)
  1 - output([t2.c2], [t2.c1]), filter(nil),
      access([t2.c2], [t2.c1]), partitions(p0)
  2 - output([t1.c2], [t1.c1]), filter(nil),
      access([t1.c2], [t1.c1]), partitions(p0)
```

- EXTENDED

```
OceanBase(admin@test)>explain extended_noaddr select * from t1,t2 where t1.c2=t2.c2 and t2.c1 > 4\G
*************************** 1. row ***************************
Query Plan: ========================================
|ID|OPERATOR   |NAME|EST. ROWS|COST   |
----------------------------------------
|0 |HASH JOIN  |    |9801000  |5933109|
|1 | TABLE SCAN|t2  |10000    |6219   |
|2 | TABLE SCAN|t1  |100000   |68478  |
========================================

Outputs & filters:
-------------------------------------
  0 - output([t1.c1], [t1.c2], [t2.c1], [t2.c2]), filter(nil),
      equal_conds([t1.c2 = t2.c2]), other_conds(nil)
  1 - output([t2.c2], [t2.c1]), filter(nil),
      access([t2.c2], [t2.c1]), partitions(p0),
      is_index_back=false,
      range_key([t2.c1]), range(4 ; MAX),
      range_cond([t2.c1 > 4])
  2 - output([t1.c2], [t1.c1]), filter(nil),
      access([t1.c2], [t1.c1]), partitions(p0),
      is_index_back=false,
      range_key([t1.__pk_increment], [t1.__pk_cluster_id], [t1.__pk_partition_id]), range(MIN,MIN,MIN
; MAX,MAX,MAX)always true
```

- TRADITIONAL format

```
OceanBase(admin@test)>explain format=TRADITIONAL select * from t1,t2 where t1.c2=t2.c2 and t2.c1 > 4\
G
*************************** 1. row ***************************
Query Plan: ======================================
|ID|OPERATOR   |NAME|EST. ROWS|COST   |
--------------------------------------
|0 |HASH JOIN  |    |9801000  |5933109|
|1 | TABLE SCAN|t2  |10000    |6219   |
|2 | TABLE SCAN|t1  |100000   |68478  |
======================================

Outputs & filters:
-------------------------------------
  0 - output([t1.c1], [t1.c2], [t2.c1], [t2.c2]), filter(nil),
      equal_conds([t1.c2 = t2.c2]), other_conds(nil)
  1 - output([t2.c2], [t2.c1]), filter(nil),
      access([t2.c2], [t2.c1]), partitions(p0)
  2 - output([t1.c2], [t1.c1]), filter(nil),
      access([t1.c2], [t1.c1]), partitions(p0)
```

- JSON format

```
OceanBase(admin@test)>explain format=JSON select * from t1,t2 where t1.c2=t2.c2 and t2.c1 > 4\G
*************************** 1. row ***************************
Query Plan: {
  "ID":2,
  "OPERATOR":"JOIN",
  "NAME":"JOIN",
  "EST.ROWS":9800999,
  "COST":5933108,
  "output": [
    "t1.c1",
    "t1.c2",
    "t2.c1",
    "t2.c2"
  ],
  "TABLE SCAN": {
    "ID":0,
    "OPERATOR":"TABLE SCAN",
    "NAME":"TABLE SCAN",
    "EST.ROWS":10000,
    "COST":6218,
    "output": [
      "t2.c2",
      "t2.c1"
    ]
  },
  "TABLE SCAN": {
    "ID":1,
    "OPERATOR":"TABLE SCAN",
    "NAME":"TABLE SCAN",
    "EST.ROWS":100000,
    "COST":68477,
    "output": [
      "t1.c2",
      "t1.c1"
    ]
  }
}
```

Each output row of EXPLAIN provides the information about a table. Each row contains the following columns:

Column name	Description
ID	The execution serial number of the plan.
OPERATOR	The executed operator.
NAME	The table that is referenced by the operator.
EST.ROWS	The estimated number of rows that are returned by the current operator.

Column name	Description
COST	The CPU time that is consumed to execute the current operator.

17.1.5.10.3.2. FLASHBACK TABLE BEFORE DROP

Description

This statement restores the deleted tables from the recycle bin.

Prerequisites

The recycle bin is enabled. You can execute `show variables like 'recyclebin';` to check whether the recycle bin is enabled.

```
obclient> show variables like 'recyclebin';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| recyclebin    | ON    |
+---------------+-------+
1 row in set (0.00 sec)
```

If the recycle bin feature is in the disabled state, you can execute `set recyclebin = on;` to enable the recycle bin. The tables in the recycle bin are not actually deleted and still occupy resources. To completely delete these tables, execute `purge recyclebin;`.

Syntax

```
FLASHBACK TABLE object_name TO BEFORE DROP [RENAME to db_name.table_name];
```

Parameter description

Parameter	Description
object_name	Specifies the name of the object or the table to be restored. The object or the table can be restored in only the database where the table resides. When you restore a table, the indexes that are related to the table are also restored.
RENAME to	Modifies the table name and the database to which the table belongs.

Examples

- Restore the deleted table t from the recycle bin.

```
obclient> create table t(id int primary key, k int);
Query OK, 0 rows affected (0.04 sec)

obclient> insert into t values(1,1);
Query OK, 1 row affected (0.00 sec)

obclient> select * from t;
+----+------+
| id | k    |
+----+------+
|  1 |    1 |
+----+------+
1 row in set (0.00 sec)

obclient>> drop table t;
Query OK, 0 rows affected (0.01 sec)

obclient> select * from t;
ORA-00942: table or view 'SYS.T' does not exist
obclient> show recyclebin;
+--------------------------------+---------------+-------+----------------------------+
| OBJECT_NAME                    | ORIGINAL_NAME | TYPE  | CREATETIME                 |
+--------------------------------+---------------+-------+----------------------------+
| __recycle_$_1_1597028971700936 | T             | TABLE | 2020-08-10 11:09:31.701033 |
+--------------------------------+---------------+-------+----------------------------+
1 row in set (0.00 sec)

obclient> flashback table t to before drop;
Query OK, 0 rows affected (0.01 sec)

obclient> select * from t;
+----+------+
| id | k    |
+----+------+
|  1 |    1 |
+----+------+
1 row in set (0.00 sec)
```

17.1.5.10.3.3. GRANT

Description

This statement is used by the system administrator to grant a user some permissions, such as object permissions and system permissions and roles.

Note

- When a current user grants an object permission, the current user must be the object owner. For example, if user1 grants user2 the SELECT permission on table t1, user1 must have the SELECT permission on table t1. The current user must also have the GRANT OPTION permission.
- When a current user grants a system permission or a role, the current user must have the permission or the role to be granted and the GRANT OPTION permission. This way, the permission or the role can be granted.
- After the user is granted the permission, the permission can take effect only when the user reconnects to ApsaraDB for OceanBase.

Syntax

```
/*Grant object permissions*/
GRANT obj_with_col_priv_list
    ON obj_clause TO grant_user_list [WITH GRANT OPTION];

obj_with_col_priv_list:
  obj_with_col_priv
  | obj_with_col_priv_list, obj_with_col_priv

obj_with_col_priv:
  obj_privilege [column_list]

obj_privilege:
    ALTER
  | AUDIT
  | COMMENT
  | DELETE
  | GRANT
  | INDEX
  | INSERT
  | LOCK
  | RENAME
  | SELECT
  | UPDATE
  | REFERENCES
  | EXECUTE
  | CREATE
  | FLASHBACK
  | READ
  | WRITE
  | DEBUG

obj_clause:
  relation_name
  | relation_name '.' relation_name
  | DIRECTORY relation_name

grant_user_list:
    grant_user [, grant_user ...]

/*Grant system permissions*/
GRANT {system_privilege_list | ALL PRIVILEGES}
    TO grantee_user [IDENTIFIED BY password] [WITH {GRANT | ADMID} OPTION];

system_privilege_list:
  system_privilege [, system_privilege ...]

system_privilege:
  CREATE SESSION
  | EXEMPT REDACTION POLICY
  | SYSDBA
  | SYSOPER
  | SYSBACKUP
  | CREATE TABLE
  | CREATE ANY TABLE
  | ALTER ANY TABLE
  | BACKUP ANY TABLE
  | DROP ANY TABLE
```

```
| LOCK ANY TABLE
| COMMENT ANY TABLE
| SELECT ANY TABLE
| INSERT ANY TABLE
| UPDATE ANY TABLE
| DELETE ANY TABLE
| FLASHBACK ANY TABLE
| CREATE ROLE
| DROP ANY ROLE
| GRANT ANY ROLE
| ALTER ANY ROLE
| AUDIT ANY
| GRANT ANY PRIVILEGE
| GRANT ANY OBJECT PRIVILEGE
| CREATE ANY INDEX
| ALTER ANY INDEX
| DROP ANY INDEX
| CREATE ANY VIEW
| DROP ANY VIEW
| CREATE VIEW
| SELECT ANY DICTIONARY
| CREATE PROCEDURE
| CREATE ANY PROCEDURE
| ALTER ANY PROCEDURE
| DROP ANY PROCEDURE
| EXECUTE ANY PROCEDURE
| CREATE SYNONYM
| CREATE ANY SYNONYM
| DROP ANY SYNONYM
| CREATE PUBLIC SYNONYM
| DROP PUBLIC SYNONYM
| CREATE SEQUENCE
| CREATE ANY SEQUENCE
| ALTER ANY SEQUENCE
| DROP ANY SEQUENCE
| SELECT ANY SEQUENCE
| CREATE TRIGGER
| CREATE ANY TRIGGER
| ALTER ANY TRIGGER
| DROP ANY TRIGGER
| CREATE PROFILE
| ALTER PROFILE
| DROP PROFILE
| CREATE USER
| ALTER USER
| DROP USER
| CREATE TYPE
| CREATE ANY TYPE
| ALTER ANY TYPE
| DROP ANY TYPE
| EXECUTE ANY TYPE
| UNDER ANY TYPE
| PURGE DBA_RECYCLEBIN
| CREATE ANY OUTLINE
| ALTER ANY OUTLINE
| DROP ANY OUTLINE
| SYSKM
| CREATE TABLESPACE
| ALTER TABLESPACE
| DROP TABLESPACE
```

```
  | SHOW PROCESS
  | ALTER SYSTEM
  | CREATE DATABASE LINK
  | CREATE PUBLIC DATABASE LINK
  | DROP DATABASE LINK
  | ALTER SESSION
  | ALTER DATABASE

/*Grant roles*/
GRANT role_list TO grantee_user [IDENTIFIED BY password] [WITH {GRANT | ADMID} OPTION];

role_list:
role [, role ...]
```

Parameter description

Parameter	Description
priv_type	Specifies the type of the permission to be granted. For more information about permission types and description, see the following table for permission type description. If you grant multiple permissions to a user at a time, separate the permission types with commas (,).
system_privilege	Specifies the type of the system permission to be granted. If you grant multiple permissions to a user at a time, separate the permission types with commas (,).
obj_clause	Specifies the level of the permission to be granted. Permissions can be divided into the following levels: • Global level: The permissions apply to all the databases. • Database level: The permissions apply to all the objects in a specified database. • Table level: The table permissions apply to all the columns in a specified table.
WITH GRANT OPTION	Specifies whether the permission can be granted to another user. When the permission is canceled, cascading is performed.
WITH ADMIN OPTION	Specifies whether the permission can be granted to another user. When the permission is canceled, cascading is not performed.

The following table describes the types of permissions that can be granted.

Table for permission type description

Permission	Description
ALL PRIVILEGES	All the permissions except the GRANT OPTION permission.
ALTER	The ALTER TABLE permission.
CREATE	The CREATE TABLE permission.
DELETE	The DELETE permission.
DROP	The DROP permission.
GRANT OPTION	The GRANT OPTION permission.
INSERT	The INSERT permission.
UPDATE	The UPDATE permission.
SELECT	The SELECT permission.
INDEX	The CREATE INDEX and DROP INDEX permissions.
SHOW VIEW	The SHOW CREATE VIEW permission.
SHOW DATABASES	The global SHOW DATABASES permission.
SUPER	The permission to execute the SET GLOBAL statement to modify global system parameters.
REFERENCES	The permission to create a constraint that refers to the table.
EXECUTE	The permission to execute the preprocessor program.
FLASHBACK	The FLASHBACK permission.
READ	The READ permission.
WRITE	The WRITE permission.

Permission	Description
CREATE SESSION	The permission to connect to the database.
EXEMPT REDACTION POLICY	The permission to bypass existing redaction policies and view data.
SYSDBA	The SYSDBA permission.
SYSOPER	The SYSOPER permission.
SYSBACKUP	The SYSBACKUP permission.
CREATE TABLE	The permission to create a table in the specified user schema.
CREATE ANY TABLE	The permission to create tables in all the user schemas except SYS.
ALTER ANY TABLE	The permission to modify tables in all the user schemas except SYS.
BACKUP ANY TABLE	The permission to create tables in all the user schemas except SYS.
DROP ANY TABLE	The permission to back up tables in all the user schemas except SYS.
LOCK ANY TABLE	The permission to lock tables in all the user schemas except SYS.
COMMENT ANY TABLE	The permission to comment tables in all the user schemas except SYS.
SELECT ANY TABLE	The permission to view tables in all the user schemas except SYS.
INSERT ANY TABLE	The permission to insert rows into tables in all the user schemas except SYS.
UPDATE ANY TABLE	The permission to update rows in tables in all the user schemas except SYS.

Permission	Description
DELETE ANY TABLE	The permission to delete tables in all the user schemas except SYS.
FLASHBACK ANY TABLE	The permission to flash back tables in all the user schemas except SYS.
CREATE ROLE	The permission to create a role.
DROP ANY ROLE	The permission to delete a role.
GRANT ANY ROLE	The permission to grant a role.
ALTER ANY ROLE	The permission to modify a role.
AUDIT ANY	The permission to modify objects in all the user schemas except SYS.
GRANT ANY PRIVILEGE	The permission to grant a system permission.
GRANT ANY OBJECT PRIVILEGE	The permission to grant an object permission.
CREATE ANY INDEX	The permission to create indexes in all the user schemas except SYS.
ALTER ANY INDEX	The permission to modify indexes in all the user schemas except SYS.
DROP ANY INDEX	The permission to delete indexes in all the user schemas except SYS.
CREATE ANY VIEW	The permission to create views in all the user schemas except SYS.
DROP ANY VIEW	The permission to delete indexes in all the user schemas except SYS.
CREATE VIEW	The permission to create a view in the specified user schema.

Permission	Description
SELECT ANY DICTIONARY	The permission to query a dictionary in the specified user schema.
CREATE PROCEDURE	The permission to create a procedure in the specified user schema.
CREATE ANY PROCEDURE	The permission to create procedures in all the user schemas except SYS.
ALTER ANY PROCEDURE	The permission to modify procedures in all the user schemas except SYS.
DROP ANY PROCEDURE	The permission to delete procedures in all the user schemas except SYS.
EXECUTE ANY PROCEDURE	The permission to perform procedures in all the user schemas except SYS.
CREATE SYNONYM	The permission to create a synonym in the specified user schema.
CREATE ANY SYNONYM	The permission to create synonyms in all the user schemas except SYS.
DROP ANY SYNONYM	The permission to delete synonyms in all the user schemas except SYS.
CREATE PUBLIC SYNONYM	The permission to create a public synonym.
DROP PUBLIC SYNONYM	The permission to delete a public synonym.
CREATE SEQUENCE	The permission to create a sequence in the specified user schema.
CREATE ANY SEQUENCE	The permission to create sequences in all the user schemas except SYS.
ALTER ANY SEQUENCE	The permission to modify sequences in all the user schemas except SYS.
DROP ANY SEQUENCE	The permission to delete sequences in all the user schemas except SYS.

Permission	Description
SELECT ANY SEQUENCE	The permission to query sequences in all the user schemas except SYS.
CREATE TRIGGER	The permission to create a trigger in the specified user schema.
CREATE ANY TRIGGER	The permission to create triggers in all the user schemas except SYS.
ALTER ANY TRIGGER	The permission to modify triggers in all the user schemas except SYS.
DROP ANY TRIGGER	The permission to delete triggers in all the user schemas except SYS.
CREATE PROFILE	The permission to create a profile.
ALTER PROFILE	The permission to modify a profile.
DROP PROFILE	The permission to delete a profile.
CREATE USER	The permission to create a user.
ALTER USER	The permission to modify a user.
DROP USER	The permission to delete a user.
CREATE TYPE	The permission to create a type in the specified user schema.
CREATE ANY TYPE	The permission to create types in all the user schemas except SYS.
ALTER ANY TYPE	The permission to modify types in all the user schemas except SYS.
DROP ANY TYPE	The permission to delete types in all the user schemas except SYS.
EXECUTE ANY TYPE	The permission to execute types in all the user schemas except SYS.

Permission	Description
UNDER ANY TYPE	The permission to create subtypes on the basis of the types in all the user schemas except SYS.
PURGE DBA_RECYCLEBIN	The permission to delete all the objects from the system recycle bin.
CREATE ANY OUTLINE	The permission to create outlines in all the user schemas except SYS.
ALTER ANY OUTLINE	The permission to modify outlines in all the user schemas except SYS.
DROP ANY OUTLINE	The permission to delete outlines in all the user schemas except SYS.
SYSKM	The SYSKM permission.
CREATE TABLESPACE	The permission to create a tablespace.
ALTER TABLESPACE	The permission to modify a tablespace.
DROP TABLESPACE	The permission to delete a tablespace.
ALTER SYSTEM	The ALTER SYSTEM permission.
CREATE DATABASE LINK	The permission to create a database link in the specified user schema.
CREATE PUBLIC DATABASE LINK	The permission to create a public database link.
DROP DATABASE LINK	The permission to delete a database link in the specified user schema.
ALTER SESSION	The permission to modify a session.
ALTER DATABASE	The permission to modify a database.

Examples

- Run the following command to grant the obsqluser01 user all the permissions.

```
OceanBase(admin@TEST)>GRANT ALL PRIVILEGES ON *. * TO obsqluser01 with grant option;
Query OK, 0 rows affected (0.03 sec)
```

17.1.5.10.3.4. KILL

Description

This statement terminates a session.

> **Note**
>
> If you have the PROCESS permission, you can view all the sessions. If you have the SUPER permission, you can terminate all the sessions and statements. Otherwise, you can view and terminate only your own sessions and statements.

Syntax

```
KILL [CONNECTION | QUERY] 'sessionid'
```

Parameter description

Parameter	Description
KILL	Terminates the specified threadid. This parameter is the same as KILL CONNECTION.
KILL CONNECTION	Terminates the specified threadid. This parameter is the same as the KILL statement that does not contain a modifier.
KILL QUERY	Terminates the statement that is being executed over the connection. The connection remains unchanged.

Examples

- Terminate the connection from the session whose sessionid is 3221638213 to the statement that is being executed, and then terminate the session.

```
OceanBase(admin@test)>show processlist;
+------------+-------+--------------------+------+---------+------+--------+------------------+
| Id         | User  | Host               | db   | Command | Time | State  | Info             |
+------------+-------+--------------------+------+---------+------+--------+------------------+
| 3221638212 | admin | 1.11.111.127:11161 | test | Query   |    0 | ACTIVE | show processlist |
| 3221638213 | admin | 1.11.111.127:11161 | test | Query   |    0 | ACTIVE | select "abcedfg" |
+------------+-------+--------------------+------+---------+------+--------+------------------+
1 row in set (0.03 sec)

OceanBase(admin@test)>kill query 3221638213;
Query OK, 0 rows affected (0.01 sec)

OceanBase(admin@test)>kill 3221638212;
Query OK, 0 rows affected (0.01 sec)
```

17.1.5.10.3.5. REVOKE

Description

This statement is used by the system administrator to revoke some permissions from a user.

The following usage description is provided:

- A user must have the permission to be revoked. For example, if user1 needs to revoke the SELECT permission on table t1 from user2, user1 must have the SELECT permission on table t1. The user must also have the GRANT OPTION permission.
- When the ALL PRIVILEGES and GRANT OPTION permissions are revoked, the current user must have the global GRANT OPTION permission or the UPDATE and DELETE permissions on the permission list.
- If the GRANT OPTION permission is unavailable, cascading is not performed for the revocation operation For example, user1 grants some permissions to user2. When the permissions are revoked from user1, the permissions are not revoked from user2. If the GRANT OPTION permission is available, cascading is performed for the revocation operation

Syntax

```
/*Revoke object permissions*/
REVOKE obj_privileges
  ON obj_clause FROM user_list;

user_list:
  user [, user ...]

obj_privileges:
  obj_privilege [, obj_privilege ...]

obj_privilege:
    ALTER
  | AUDIT
  | COMMENT
  | DELETE
  | GRANT
  | INDEX
  | INSERT
  | LOCK
  | RENAME
  | SELECT
  | UPDATE
  | REFERENCES
```

```
  | EXECUTE
  | CREATE
  | FLASHBACK
  | READ
  | WRITE
  | DEBUG

obj_clause:
  relation_name
  | relation_name '.' relation_name
  | DIRECTORY relation_name

relation_name:
  STR_VALUE

/*Revoke system permissions*/
REVOKE {system_privilege_list | ALL PRIVILEGES}
  FROM user_list;
REVOKE ALL [PRIVILEGES], GRANT_OPTION FROM user_list;

system_privilege_list:
  system_privilege [, system_privilege ...]

system_privilege:
  CREATE SESSION
  | EXEMPT REDACTION POLICY
  | SYSDBA
  | SYSOPER
  | SYSBACKUP
  | CREATE TABLE
  | CREATE ANY TABLE
  | ALTER ANY TABLE
  | BACKUP ANY TABLE
  | DROP ANY TABLE
  | LOCK ANY TABLE
  | COMMENT ANY TABLE
  | SELECT ANY TABLE
  | INSERT ANY TABLE
  | UPDATE ANY TABLE
  | DELETE ANY TABLE
  | FLASHBACK ANY TABLE
  | CREATE ROLE
  | DROP ANY ROLE
  | GRANT ANY ROLE
  | ALTER ANY ROLE
  | AUDIT ANY
  | GRANT ANY PRIVILEGE
  | GRANT ANY OBJECT PRIVILEGE
  | CREATE ANY INDEX
  | ALTER ANY INDEX
  | DROP ANY INDEX
  | CREATE ANY VIEW
  | DROP ANY VIEW
  | CREATE VIEW
  | SELECT ANY DICTIONARY
  | CREATE PROCEDURE
  | CREATE ANY PROCEDURE
  | ALTER ANY PROCEDURE
  | DROP ANY PROCEDURE
```

```
  | EXECUTE ANY PROCEDURE
  | CREATE SYNONYM
  | CREATE ANY SYNONYM
  | DROP ANY SYNONYM
  | CREATE PUBLIC SYNONYM
  | DROP PUBLIC SYNONYM
  | CREATE SEQUENCE
  | CREATE ANY SEQUENCE
  | ALTER ANY SEQUENCE
  | DROP ANY SEQUENCE
  | SELECT ANY SEQUENCE
  | CREATE TRIGGER
  | CREATE ANY TRIGGER
  | ALTER ANY TRIGGER
  | DROP ANY TRIGGER
  | CREATE PROFILE
  | ALTER PROFILE
  | DROP PROFILE
  | CREATE USER
  | ALTER USER
  | DROP USER
  | CREATE TYPE
  | CREATE ANY TYPE
  | ALTER ANY TYPE
  | DROP ANY TYPE
  | EXECUTE ANY TYPE
  | UNDER ANY TYPE
  | PURGE DBA_RECYCLEBIN
  | CREATE ANY OUTLINE
  | ALTER ANY OUTLINE
  | DROP ANY OUTLINE
  | SYSKM
  | CREATE TABLESPACE
  | ALTER TABLESPACE
  | DROP TABLESPACE
  | SHOW PROCESS
  | ALTER SYSTEM
  | CREATE DATABASE LINK
  | CREATE PUBLIC DATABASE LINK
  | DROP DATABASE LINK
  | ALTER SESSION
  | ALTER DATABASE

/*Revoke roles*/
REVOKE role_list FROM user;

role_list:
role [, role ...]
```

Parameter description

Parameter	Description

Parameter	Description
obj_privileges	Specifies the type of the object permission to be revoked. For more information about permission types and description, see the following table for permission type description. If you revoke multiple permissions at a time, separate the permission types with commas (,).
system_privilege	Specifies the type of the system permission to be revoked. If you revoke multiple permissions at a time, separate the permission types with commas (,).
obj_clause	Specifies the level of the permission to be revoked. relation_name specifies the name of the specific object. Permissions can be divided into the following levels: • Global level: The permissions apply to all the databases. • Database level: The permissions apply to all the objects in a specified database. • Table level: The permissions apply to all the columns in a specified table.

The following table describes the types of permissions that can be revoked.

Table for permission type description

Permission	Description
ALL PRIVILEGES	All the permissions except the GRANT OPTION permission.
ALTER	The ALTER TABLE permission.
CREATE	The CREATE TABLE permission.
DELETE	The DELETE permission.
DROP	The DROP permission.
GRANT OPTION	The GRANT OPTION permission.
INSERT	The INSERT permission.
UPDATE	The UPDATE permission.

Permission	Description
SELECT	The SELECT permission.
INDEX	The CREATE INDEX and DROP INDEX permissions.
SHOW VIEW	The SHOW CREATE VIEW permission.
SHOW DATABASES	The global SHOW DATABASES permission.
SUPER	The permission to execute the SET GLOBAL statement to modify global system parameters.
REFERENCES	The permission to create a constraint that refers to the table.
EXECUTE	The permission to execute the preprocessor program.
FLASHBACK	The FLASHBACK permission.
READ	The READ permission.
WRITE	The WRITE permission.
CREATE SESSION	The permission to connect to the database.
EXEMPT REDACTION POLICY	The permission to bypass existing redaction policies and view data.
SYSDBA	The SYSDBA permission.
SYSOPER	The SYSOPER permission.
SYSBACKUP	The SYSBACKUP permission.
CREATE TABLE	The permission to create a table in the specified user schema.
CREATE ANY TABLE	The permission to create tables in all the user schemas except SYS.

Permission	Description
ALTER ANY TABLE	The permission to modify tables in all the user schemas except SYS.
BACKUP ANY TABLE	The permission to create tables in all the user schemas except SYS.
DROP ANY TABLE	The permission to back up tables in all the user schemas except SYS.
LOCK ANY TABLE	The permission to lock tables in all the user schemas except SYS.
COMMENT ANY TABLE	The permission to comment tables in all the user schemas except SYS.
SELECT ANY TABLE	The permission to view tables in all the user schemas except SYS.
INSERT ANY TABLE	The permission to insert rows into tables in all the user schemas except SYS.
UPDATE ANY TABLE	The permission to update rows in tables in all the user schemas except SYS.
DELETE ANY TABLE	The permission to delete tables in all the user schemas except SYS.
FLASHBACK ANY TABLE	The permission to flash back tables in all the user schemas except SYS.
CREATE ROLE	The permission to create a role.
DROP ANY ROLE	The permission to delete a role.
GRANT ANY ROLE	The permission to grant a role.
ALTER ANY ROLE	The permission to modify a role.
AUDIT ANY	The permission to modify objects in all the user schemas except SYS.

Permission	Description
GRANT ANY PRIVILEGE	The permission to grant a system permission.
GRANT ANY OBJECT PRIVILEGE	The permission to grant an object permission.
CREATE ANY INDEX	The permission to create indexes in all the user schemas except SYS.
ALTER ANY INDEX	The permission to modify indexes in all the user schemas except SYS.
DROP ANY INDEX	The permission to delete indexes in all the user schemas except SYS.
CREATE ANY VIEW	The permission to create views in all the user schemas except SYS.
DROP ANY VIEW	The permission to delete indexes in all the user schemas except SYS.
CREATE VIEW	The permission to create a view in the specified user schema.
SELECT ANY DICTIONARY	The permission to query a dictionary in the specified user schema.
CREATE PROCEDURE	The permission to create a procedure in the specified user schema.
CREATE ANY PROCEDURE	The permission to create procedures in all the user schemas except SYS.
ALTER ANY PROCEDURE	The permission to modify procedures in all the user schemas except SYS.
DROP ANY PROCEDURE	The permission to delete procedures in all the user schemas except SYS.
EXECUTE ANY PROCEDURE	The permission to perform procedures in all the user schemas except SYS.
CREATE SYNONYM	The permission to create a synonym in the specified user schema.

Permission	Description
CREATE ANY SYNONYM	The permission to create synonyms in all the user schemas except SYS.
DROP ANY SYNONYM	The permission to delete synonyms in all the user schemas except SYS.
CREATE PUBLIC SYNONYM	The permission to create a public synonym.
DROP PUBLIC SYNONYM	The permission to delete a public synonym.
CREATE SEQUENCE	The permission to create a sequence in the specified user schema.
CREATE ANY SEQUENCE	The permission to create sequences in all the user schemas except SYS.
ALTER ANY SEQUENCE	The permission to modify sequences in all the user schemas except SYS.
DROP ANY SEQUENCE	The permission to delete sequences in all the user schemas except SYS.
SELECT ANY SEQUENCE	The permission to query sequences in all the user schemas except SYS.
CREATE TRIGGER	The permission to create a trigger in the specified user schema.
CREATE ANY TRIGGER	The permission to create triggers in all the user schemas except SYS.
ALTER ANY TRIGGER	The permission to modify triggers in all the user schemas except SYS.
DROP ANY TRIGGER	The permission to delete triggers in all the user schemas except SYS.
CREATE PROFILE	The permission to create a profile.
ALTER PROFILE	The permission to modify a profile.

Permission	Description
DROP PROFILE	The permission to delete a profile.
CREATE USER	The permission to create a user.
ALTER USER	The permission to modify a user.
DROP USER	The permission to delete a user.
CREATE TYPE	The permission to create a type in the specified user schema.
CREATE ANY TYPE	The permission to create types in all the user schemas except SYS.
ALTER ANY TYPE	The permission to modify types in all the user schemas except SYS.
DROP ANY TYPE	The permission to delete types in all the user schemas except SYS.
EXECUTE ANY TYPE	The permission to execute types in all the user schemas except SYS.
UNDER ANY TYPE	The permission to create subtypes on the basis of the types in all the user schemas except SYS.
PURGE DBA_RECYCLEBIN	The permission to delete all the objects from the system recycle bin.
CREATE ANY OUTLINE	The permission to create outlines in all the user schemas except SYS.
ALTER ANY OUTLINE	The permission to modify outlines in all the user schemas except SYS.
DROP ANY OUTLINE	The permission to delete outlines in all the user schemas except SYS.
SYSKM	The SYSKM permission.

Permission	Description
CREATE TABLESPACE	The permission to create a tablespace.
ALTER TABLESPACE	The permission to modify a tablespace.
DROP TABLESPACE	The permission to delete a tablespace.
ALTER SYSTEM	The ALTER SYSTEM permission.
CREATE DATABASE LINK	The permission to create a database link in the specified user schema.
CREATE PUBLIC DATABASE LINK	The permission to create a public database link.
DROP DATABASE LINK	The permission to delete a database link in the specified user schema.
ALTER SESSION	The permission to modify a session.
ALTER DATABASE	The permission to modify a database.

Examples

Run the following command to revoke all the permissions from the obsqluser01 user:

```
OceanBase(admin@TEST)>REVOKE ALL PRIVILEGES FROM sqluser;
Query OK, 0 rows affected (0.10 sec)
```

17.1.5.10.3.6. SAVEPOINT

Description

The SAVEPOINT statement performs a partial rollback of a transaction.

Syntax

1. Create a savepoint:

```
SAVEPOINT spname
```

2. Roll back to a savepoint:

```
ROLLBACK [WORK] to [SAVEPOINT] spname
```

3. Delete a savepoint:

```
RELEASE SAVEPOINT spname
```

Parameter description

Parameter	Description
spname	Specifies the name of the savepoint. Savepoints are unique in the transaction scope. A savepoint overwrites the preceding savepoint that has the same name as the save point. After you create a savepoint, you can roll back the transaction to the specified savepoint. You can also use the `ROLLBACK` statement to roll back the entire transaction.

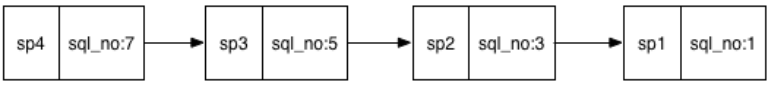
Examples

Assume that the following statements are executed in a transaction.

sql_no	Statement	Partition
1	update...	p1, p4
	savepoint sp1	
2	update...	p2, p4
3	update...	p3, p5
	savepoint sp2	
4	update...	p1, p3, p6
5	update...	p1, p5
	savepoint sp3	
6	select...	
7	update...	p5, p6
	savepoint sp4	

Record savepoints

You can create savepoints before you submit transactions. You must connect the transaction savepoints to a linked list by using the order in which they are created. The preceding transaction contains seven SQL statements and four savepoints. The linked list of savepoints is shown in the following figure where each node records a <spname, sql_no> mapping.



Transaction participant list

To roll back all the modifications that are made after an SQL statement in a transaction, the participants and sql_no that each statement involves are recorded. The preceding transaction contains seven SQL statements and six partitions that range from p1 to p6:

p1	p2	p3	p4	p5	p6
1,4,5	2	2,3,4	1,2	3,5,7	4,7

Savepoint rollback process

1. Query the sql_no that corresponds to spname by using the savepoint linked list

For example, the user executes the `ROLLBACK to SAVEPOINT sp2` statement. Based on the savepoint linked list, the queries sql_no that corresponds to sp2 is 3.

2. Query the partition that corresponds to sql_no by using the transaction participant list

Based on the transaction participant list, the partitions whose sql_no is larger than 3 involve p1, p3, p5, and p6.

3. Roll back the data of the partitions

The scheduler initiates a rollback request to these partitions that are queried in Step 2. This request rolls back all modifications that are made by the transaction after sp2 on these partitions. Some modifications that are related to this transaction on p1, p3, and p5 are rolled back, and all the modifications that are related to this transaction on p6 are rolled back.

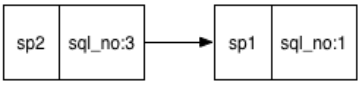
4. Update the information about the transaction participant list

Modify the transaction participant list and delete the operation information in which sql_no is larger than 3 from the transaction participant list. p6 can be deleted from the participant list because all the modifications on p6 are rolled back.

p1	p2	p3	p4	p5	p6
1	2	2, 3	1, 2	3	

5. Delete invalid savepoints

After the user executes the `ROLLBACK to SAVEPOINT sp2` statement, the system deletes the savepoints sp3 and sp4. Then, the transaction cannot be rolled back to sp3 and sp4.



17.1.5.10.3.7. SET NAMES

Description

This statement sets the character set encoding for the current connected database.

Syntax

```
SET NAMES 'charset_name' [COLLATE 'collation_name']
```

Parameter description

Parameter	Description
charset_name	Specifies a character set.
collation_name	Specifies a collation for the character set. If you do not specify this parameter, the default collation for the character set is used.

> **Note**
>
> This statement specifies three session system variables `character_set_client`, `character_set_connection`, and `character_set_results` as the specified character set. This statement specifies `character_set_connection` as `charset_name`, and `collation_connection` as the default collation of `charset_name`.

Examples

- Specify the character set and collation of the database.

```
OceanBase(admin@test)>SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci';
Query OK, 0 rows affected (0.00 sec)

OceanBase(admin@test)>show variables like 'character_set_c%';
+--------------------------+---------+
| Variable_name            | Value   |
+--------------------------+---------+
| character_set_client     | utf8mb4 |
| character_set_connection | utf8mb4 |
+--------------------------+---------+
2 rows in set (0.01 sec)

OceanBase(admin@test)>show variables like 'collation_connection';
+----------------------+--------------------+
| Variable_name        | Value              |
+----------------------+--------------------+
| collation_connection | utf8mb4_general_ci |
+----------------------+--------------------+
1 row in set (0.05 sec)


OceanBase(admin@test)>SET NAMES 'gbk';
Query OK, 0 rows affected (0.02 sec)

OceanBase(admin@test)>show variables like 'character_set_c%';
+--------------------------+-------+
| Variable_name            | Value |
+--------------------------+-------+
| character_set_client     | gbk   |
| character_set_connection | gbk   |
+--------------------------+-------+
2 rows in set (0.01 sec)


OceanBase(admin@test)>show variables like 'collation_connection';
+----------------------+----------------+
| Variable_name        | Value          |
+----------------------+----------------+
| collation_connection | gbk_chinese_ci |
+----------------------+----------------+
1 row in set (0.01 sec)
```

17.1.5.10.3.8. SET PASSWORD

Description

This statement changes the password of the current logon user or other users of ApsaraDB for OceanBase.

Syntax

```
SET PASSWORD [FOR user] = PASSWORD(password);
ALTER USER user IDENTIFIED BY password;

password:
  STR_VALUE
```

Parameter description

Parameter	Description
FOR user	If you do not specify the FOR user clause, the system changes the password for the current user. A user who logs on to ApsaraDB for OceanBase can change the password. If you specify the FOR user clause, the system changes the password for the specified user. To change the password for a specified user, you must have the CREATE USER system permission.

Examples

- Run the following command to change the password for the sqluser user to abc123:

```
oceanBase(admin@TEST)>SET PASSWORD for sqluser = PASSWORD(abc123);
Query OK, 0 rows affected (0.02 sec)
```

17.1.5.10.3.9. SET VARIABLE

Description

The SET VARIABLE statement is used to specify user-defined and system session variables.

Syntax

```
SET var1_name = var1_value, var2_name = var2_value, ... ;
```

Parameter description

Parameter	Description
var1_name	Specifies the variable to be specified. It can be a user-defined variable or a system variable. To set multiple variables, separate them with commas (,).

Examples

```
OceanBase(TEST@TEST)>set @a = 1, @b = 2, @c = 3;
Query OK, 0 rows affected (0.02 sec)

OceanBase(TEST@TEST)>set @@tx_isolation = 'read-what';
ORA-00600: internal error code, arguments: -5145, Variable 'tx_isolation' can't be set to the value o
f 'read-what'

OceanBase(TEST@TEST)>set @@tx_isolation = 'read-committed';
Query OK, 0 rows affected (0.01 sec)

OceanBase(TEST@TEST)>select @a,@b,@c,@@tx_isolation from dual;
+------+------+------+----------------+
| @A   | @B   | @C   | @@TX_ISOLATION |
+------+------+------+----------------+
|    1 |    2 |    3 | read-committed |
+------+------+------+----------------+
1 row in set (0.01 sec)
```

17.1.5.10.3.10. SHOW

Description

`show parameters` is used to show system variables.

`show errors` is used to show the error message of the current object.

Syntax

```
show parameters;
show errors;
```

Parameter description

None

Examples

- Show all system variables.

```
show parameters;
```

17.1.5.10.3.11. SHOW RECYCLEBIN

Description

The SHOW RECYCLEBIN statement is used to view content in the recycle bin.

Syntax

```
 SHOW RECYCLEBIN;
```

Parameter description

None

Examples

- View the content in the recycle bin.

```
OceanBase(admin@test)> create table t1(c1 int);
Query OK, 0 rows affected (0.24 sec)

OceanBase(admin@test)> drop table t1;
Query OK, 0 rows affected (0.07 sec)

OceanBase(admin@test)> show recyclebin;
+-----------------------------------------+---------------+-------+----------------------------+
| OBJECT_NAME                             | ORIGINAL_NAME | TYPE  | CREATETIME                 |
+-----------------------------------------+---------------+-------+----------------------------+
| __recycle_$_1_1099511628776_1099511677777 | t1            | TABLE | 2017-10-20 17:27:40.881506 |
+-----------------------------------------+---------------+-------+----------------------------+
1 row in set (0.02 sec)
```

17.1.5.10.3.12. SHRINK

Description

This statement reorganizes the data in the table. This indicates that the data is rewritten during the merge process.

Syntax

```
ALTER TABLE table_name SHRINK SPACE
```

Parameter description

Parameter	Description
table_name	The table name.

Examples

- Reorganize the data of table t1.

```
OceanBase(admin@test)>ALTER TABLE t1 SHRINK SPACE;
```

17.1.5.10.3.13. TRANSACTION

Description

The TRANSACTION statement starts a transaction.

A database transaction refers to a series of operations that are executed as a single logical unit of work. Transaction processing can be used to maintain the integrity of databases because it ensures that SQL operations in a batch are all executed or are not executed at all.

Explicit transactions are user-defined or user-specified transactions. The transactions explicitly start with a BEGIN TRANSACTION statement or BEGIN and BEGIN WORK statements (supported as the aliases of START TRANSACTION), and explicitly end with a COMMIT or ROLLBACK statement.

Syntax

```
transaction_stmt:
      START TRANSACTION [READ ONLY | READ WRITE];
    | BEGIN [WORK];
    | COMMIT [WORK];
    | ROLLBACK [WORK];
    | SET TRANSACTION {READ ONLY | READ WRITE};
```

Parameter description

Parameter	Description
START TRANSACTION [READ ONLY | READ WRITE]	The statement that is used to start a transaction. After the transaction is started, the subsequent SQL data manipulation language (DML) statements, such as INSERT, UPDATE, and DELETE, take effect only when the transaction is explicitly committed. The READ ONLY clause indicates that the transaction is started in READ ONLY mode. You cannot perform modifications within a transaction. The READ WRITE clause indicates that the transaction is started in READ WRITE mode. It is the default mode.
BEGIN	BEGIN and BEGIN WORK are supported as the aliases of START TRANSACTION.
COMMIT	Commits the current transaction.
ROLLBACK	Rolls back the current transaction.
SET TRANSACTION {READ ONLY | READ WRITE}	Sets the mode of the current transaction to READ ONLY or READ WRITE.

Examples

Assume that the following table a is available.

id	name	num	sell_date
1	a	100	2013-06-21 10:06:43
2	b	200	2013-06-21 13:07:21
3	a	50	2013-06-21 13:08:15

1. Run the following commands in sequence to start to execute the transaction, change the name of the row whose id is 3 to c, and insert a row of sales records of a.

```
OceanBase(admin@test)> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

OceanBase(admin@test)> UPDATE a SET name = 'c' WHERE id = 3;
Query OK, 1 rows affected (0.00 sec)

OceanBase(admin@test)> INSERT INTO a VALUES (4, 'a', 30, '2013-06-21 16:09:13');
Query OK, 1 rows affected (0.00 sec)

OceanBase(admin@test)> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

2. After you commit the transaction, run the following command to query information about table a:

```
SELECT * FROM a;
```

The following result is returned:

id	name	num	sell_date
1	a	100	2013-06-21 10:06:43
2	b	200	2013-06-21 13:07:21
3	c	50	2013-06-21 13:08:15
4	a	30	2013-06-21 16:09:13

Notice

Before the transaction is committed, you can check whether the operations in the current transaction have taken effect. For example, you can insert `SELECT * FROM a;` before the COMMIT keyword. The latest result can be retrieved for the access from the session of the current transaction. For the access from outside the session of the current transaction, the result does not take effect. Before the transaction is committed, your previous operations are invisible except the connection to the current transaction. To roll back the transaction, replace `COMMIT` with `ROLLBACK`.

18.Data Transmission Service (DTS)

18.1. User Guide

18.1.1. What is DTS?

Data Transmission Service (DTS) is a data service that is provided by Alibaba Cloud. DTS supports data transmission between various types of data sources, such as relational databases and big data systems.

Features

DTS has the following advantages over traditional data migration and synchronization tools: high compatibility, high performance, security, reliability, and ease of use. DTS allows you to simplify data transmission and focus on business development.

Feature	Description
Data migration	You can use DTS to migrate data between homogeneous and heterogeneous data sources. This feature applies to the following scenarios: data migration to Alibaba Cloud, data migration between instances within Alibaba Cloud, and database splitting and scale-out.
Data synchronization	You can use DTS to synchronize data between data sources. This feature applies to the following scenarios: disaster recovery, data backup, load balancing, cloud BI systems, and real-time data warehousing.
Change tracking	You can use DTS to track data changes from user-created MySQL databases, ApsaraDB RDS for MySQL instances, Cloud Native Distributed Database PolarDB-X instances (formerly known as DRDS), and user-created Oracle databases in real time. This feature applies to the following scenarios: cache updates, business decoupling, asynchronous data processing, synchronization of heterogeneous data, and synchronization of extract, transform, and load (ETL) operations.

18.1.2. Log on to the DTS console

This topic describes how to log on to the Data Transmission Service (DTS) console. Google Chrome is used in this example.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

> **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
>
> - Uppercase or lowercase letters
> - Digits
> - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products > Data Transmission Service**.
5. Select an **organization** and **region**, and then click **DTS**.

18.1.3. Data migration

18.1.3.1. Supported databases and migration types

You can use Data Transmission Service (DTS) to migrate data between homogeneous and heterogeneous data sources. Typical scenarios include data migration to Alibaba Cloud, data migration between instances within Alibaba Cloud, and database splitting and scale-out. This topic describes the database types, database versions, or migration types that are supported by the data migration feature.

Source database	Destination database	Migration type
• User-created MySQL database Version 5.1, 5.5, 5.6, 5.7, or 8.0 • ApsaraDB RDS for MySQL All versions	User-created MySQL database Version 5.1, 5.5, 5.6, 5.7, or 8.0	• Schema migration • Full data migration • Incremental data migration
	ApsaraDB RDS for MySQL All versions	• Schema migration • Full data migration • Incremental data migration
	PolarDB-X (formerly known as DRDS) All versions	• Full data migration • Incremental data migration
	User-created Oracle database (RAC or non-RAC architecture) Version 9i, 10g, 11g, 12c, 18c, or 19c	• Full data migration • Incremental data migration

Source database	Destination database	Migration type
	User-created Kafka database Versions 0.1 to 2.0	• Schema migration • Full data migration • Incremental data migration
User-created SQL Server database Version 2005, 2008, 2008 R2, 2012, 2014, 2016, or 2017 **Note** • DTS does not support SQL Server clusters or SQL Server Always On availability groups (AOAGs). • If the version of the source database is 2005, incremental data migration is not supported.	• User-created SQL Server database Version 2005, 2008, 2008 R2, 2012, 2014, 2016, or 2017 **Note** DTS does not support SQL Server clusters or SQL Server Always On availability groups (AOAGs). • ApsaraDB RDS for SQL Server Version 2008, 2008 R2, 2012, 2014, 2016, or 2017	• Schema migration • Full data migration • Incremental data migration
User-created Oracle database (RAC or non-RAC architecture) Version 9i, 10g, 11g, 12c, 18c, or 19c	User-created Oracle database (RAC or non-RAC architecture) Version 9i, 10g, 11g, 12c, 18c, or 19c	• Schema migration • Full data migration • Incremental data migration
	PolarDB Version 9.3, 9.6, 10, or 11	• Schema migration • Full data migration • Incremental data migration
	User-created MySQL database Version 5.1, 5.5, 5.6, 5.7, or 8.0	• Schema migration • Full data migration • Incremental data migration
	ApsaraDB RDS for MySQL All versions	• Schema migration • Full data migration • Incremental data migration

Source database	Destination database	Migration type
	PolarDB-X All versions	• Full data migration • Incremental data migration
	AnalyticDB for MySQL Version 2.0 or 3.0	• Schema migration • Full data migration • Incremental data migration
• User-created PostgreSQL database Version 9.4, 9.5, 9.6, or 10.x • ApsaraDB RDS for PostgreSQL Version 9.4 or 10	• User-created PostgreSQL database Version 9.4, 9.5, 9.6, or 10.x • ApsaraDB RDS for PostgreSQL Version 9.4 or 10	• Schema migration • Full data migration • Incremental data migration
PolarDB Version 9.3, 9.6, 10, or 11	User-created Kafka database Versions 0.1 to 2.0	Incremental data migration
	User-created Oracle database (RAC or non-RAC architecture) Version 9i, 10g, 11g, 12c, 18c, or 19c	• Full data migration • Incremental data migration
	PolarDB Version 9.3, 9.6, 10, or 11	• Schema migration • Full data migration • Incremental data migration
User-created Redis database Version 2.8, 3.0, 3.2, 4.0, or 5.0	User-created Redis database Version 2.8, 3.0, 3.2, 4.0, or 5.0	• Full data migration • Incremental data migration
User-created MongoDB database Version 3.0, 3.2, 3.4, 3.6, 4.0 or 4.2	User-created MongoDB database Version 3.0, 3.2, 3.4, 3.6, 4.0 or 4.2	• Full data migration • Incremental data migration

18.1.3.2. Create a data migration instance

Before you configure a task to migrate data, you must create a data migration instance. This topic describes how to create a data migration instance in the Data Transmission Service (DTS) console.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. In the upper-right corner, click **Create Migration Task**.
4. In the Create DTS Instances dialog box, select a region, and enter the number of data migration instances that you want to create.

   > **Note** In the Create DTS Instances dialog box, you can view the total number of instances, the number of existing instances, and the number of instances that can be created.

5. Click **Create**.

18.1.3.3. Configure data migration tasks

18.1.3.3.1. Migrate data from a user-created MySQL database to an ApsaraDB RDS for MySQL instance

This topic describes how to migrate data from a user-created MySQL database to an ApsaraDB RDS for MySQL instance by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. You can select all of the supported migration types to ensure service continuity.

Prerequisites

- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- If you need to perform incremental data migration, binary logging must be enabled for the user-created MySQL database and the following requirements must be met:
  - The value of the binlog_format parameter is set to row.
  - The value of the binlog_row_image parameter is set to full.

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- DTS uses the `ROUND(COLUMN,PRECISION)` function to retrieve values from columns of the FLOAT or DOUBLE data type. If you do not specify a precision, DTS sets the precision for the FLOAT data type to 38 digits and the precision for the DOUBLE data type to 308 digits. You must check whether the precision settings meet your business requirements.
- DTS automatically creates a destination database in the ApsaraDB RDS for MySQL instance. However, if the name of the source database is invalid, you must manually create a database in the ApsaraDB RDS for MySQL instance before you configure the data migration task.
- If you perform a primary/secondary switchover on the source database when the data migration task is running, the task fails.
- Migration latency is the difference between the timestamp of the latest migrated data in the destination database and the current timestamp in the source database. If no data manipulation language (DML) operations are performed on the source database for a long time, the migration latency displayed in the DTS console may be inaccurate. If the latency of the migration task is too high, you can perform a DML operation on the source database to update the latency.

> **Note** If you select an entire database as the object to be migrated, you can create a heartbeat table. The heartbeat table is updated or receives data every second.

Migration types

- Schema migration

  DTS migrates the schemas of the required objects to the destination instance. DTS supports schema migration for the following types of objects: table, view, trigger, stored procedure, and function.

  > **Note**
  > - During schema migration, DTS changes the value of the SECURITY attribute in views, stored procedures, and functions from DEFINER to INVOKER.
  > - DTS does not migrate user information. Before a user can call views, stored procedures, and functions of the destination database, you must grant the read/write permissions to the user.

- Full data migration

  DTS migrates historical data of the required objects from the user-created MySQL database to the destination database in the ApsaraDB RDS for MySQL instance.

  > **Note** During full data migration, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After full data migration is complete, the tablespace of the destination instance is larger than that of the source database.

- Incremental data migration

  After full data migration is complete, DTS retrieves binary log files from the user-created MySQL database. Then, DTS synchronizes incremental data from the user-created MySQL database to the destination ApsaraDB RDS for MySQL instance. Incremental data migration allows you to ensure service continuity when you migrate data from a user-created MySQL database to Alibaba Cloud.

Procedure

1. Create a data migration instance.
2. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
3. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.
	Database Type	Select **MySQL**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created MySQL database.

Source Section Database	Parameter	Description
	Port Number	Enter the service port number of the user-created MySQL database. The default port number is **3306**.
	Database Account	Enter the account of the user-created MySQL database. If you need to migrate incremental data, the account must have the SELECT permission on the objects to be migrated, the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, and the SHOW VIEW permission. If you do not need to migrate incremental data, the account must have the SELECT permission on the objects to be migrated.
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select **RDS Instance**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	RDS Instance ID	Select the ID of the destination RDS instance.
	Database Account	Enter the database account of the destination RDS instance. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. If a whitelist is configured for the user-created MySQL database, you must manually add the CIDR blocks of DTS servers to the whitelist of the database. To obtain the CIDR blocks of DTS servers, click **Copy to Clipboard** in the dialog box that appears. After you add the CIDR blocks of DTS servers to the whitelist of the database, click **Next**.

   > **Note** If you do not need to configure a whitelist for the user-created MySQL database, ignore the preceding settings and click **Next**.

6. Select the migration types and the objects to be migrated.

Setting	Description
Select the migration types	◦ To perform only full data migration, select **Schema Migration** and **Full Data Migration**. ◦ To ensure service continuity during data migration, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.

Setting	Description
Select the objects to be migrated	Select objects from the **Available** section and click the icon to move the objects to the **Selected** section. **Note** ◦ You can select columns, tables, or databases as the objects to be migrated. ◦ By default, after an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. ◦ If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.
Specify the retry time for failed connection to the source or destination database	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 12 hours. You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data migration task. Otherwise, the data migration task fails. **Note** When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

7. In the lower-right corner of the page, click **Precheck**.

   Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

8. After the task passes the precheck, click **Next**.

   Note If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.2. Migrate data from a user-created MySQL database to a PolarDB-X instance

This topic describes how to migrate data from a user-created MySQL database to a PolarDB-X instance by using Data Transmission Service (DTS). PolarDB-X is formerly known as Distributed Relational Database Service (DRDS).

Precautions

- DTS cannot migrate schemas from a user-created MySQL database to a PolarDB-X instance.

  Note During schema migration, DTS migrates the schemas of the required objects, such as tables, from the source database to the destination database.

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- DTS uses the `ROUND(COLUMN,PRECISION)` function to retrieve values from columns of the FLOAT or DOUBLE data type. If you do not specify a precision, DTS sets the precision for the FLOAT data type to 38 digits and the precision for the DOUBLE data type to 308 digits. You must check whether the precision settings meet your business requirements.
- If you perform a primary/secondary switchover on the source database when the data migration task is running, the task fails.
- Migration latency is the difference between the timestamp of the latest migrated data in the destination database and the current timestamp in the source database. If no data manipulation language (DML) operations are performed on the source database for a long time, the migration latency displayed in the DTS console may be inaccurate. If the latency of the migration task is too high, you can perform a DML operation on the source database to update the latency.

> **Note** If you select an entire database as the object to be migrated, you can create a heartbeat table. The heartbeat table is updated or receives data every second.

SQL operations that can be synchronized during incremental data migration

INSERT, UPDATE, DELETE, and REPLACE

Permissions required for database accounts

Database	Full data migration	Incremental data migration
User-created MySQL database	The SELECT permission	The REPLICATION SLAVE, REPLICATION CLIENT, SHOW VIEW, and SELECT permissions
PolarDB-X	The read and write permissions	The read and write permissions

Procedure

1. Create databases and tables in the destination PolarDB-X instance based on the schemas of the source tables.
2. Create a data migration instance.
3. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
4. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
	Instance Type	Select **User-Created Database with Public IP Address**.

Section	Parameter	Description
Source Database	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.
	Database Type	Select **MySQL**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created MySQL database.
	Port Number	Enter the service port number of the user-created MySQL database. The default port number is **3306**.
	Database Account	Enter the account of the user-created MySQL database. For more information about the permissions that are required for the account, see Permissions required for database accounts.
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select **DRDS Instance**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the PolarDB-X instance.
	Database Name	Enter the name of the destination database.
	Database Account	Enter the database account of the PolarDB-X instance. For more information about the permissions that are required for the account, see Permissions required for database accounts.
	Database Password	Enter the password of the destination database account.

5. In the lower-right corner of the page, click **Set Whitelist and Next**.
6. If a whitelist is configured for the user-created MySQL database, you must manually add the CIDR blocks of DTS servers to the whitelist of the database. To obtain the CIDR blocks of DTS servers, click **Copy to Clipboard** in the dialog box that appears. After you add the CIDR blocks of DTS servers to the whitelist of the database, click **Next**.

   Note If you do not need to configure a whitelist for the user-created MySQL database, ignore the preceding settings and click **Next**.

7. Select the migration types and the objects to be migrated.

Setting	Description
Select the migration types	○ To perform only full data migration, select only **Full Data Migration**. ○ To ensure service continuity during data migration, select both **Full Data Migration** and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.

Setting	Description
Select the objects to be migrated	Select one or more objects from the **Available** section and click the icon to move the objects to the **Selected** section. **Note** • You can select columns, tables, or databases as the objects to be migrated. • By default, after an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. • If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.
Specify the retry time for failed connection to the source or destination database	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 12 hours. You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data migration task. Otherwise, the data migration task fails. **Note** When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

8. In the lower-right corner of the page, click **Precheck**.

   Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

9. After the task passes the precheck, click **Next**.

   Note If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.3. Migrate data from an ApsaraDB RDS for MySQL instance to a user-created Oracle database

This topic describes how to migrate data from an ApsaraDB RDS for MySQL instance to a user-created Oracle database by using Data Transmission Service (DTS). DTS supports full data migration and incremental data migration. You can select these migration types to ensure service continuity of the source ApsaraDB RDS for MySQL instance during data migration.

Prerequisites

The destination Oracle database is created. The schema of the Oracle database is the same as the schema of the source database in the ApsaraDB RDS for MySQL instance. This is because DTS does not support schema migration from an ApsaraDB RDS for MySQL instance to a user-created Oracle database.

Permissions required for database accounts

Database\Migration type	Full data migration	Incremental data migration
Source ApsaraDB RDS for MySQL instance	The read and write permissions	The read and write permissions
Destination Oracle database	The read and write permissions	The read and write permissions

Procedure

1. Create a data migration instance.
2. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
3. Optional. Enter a name for the task.

   DTS automatically generates a name for each task. Duplicate task names are allowed. You can edit the task name based on your needs. We recommend that you specify an informative name for easy identification.
4. Configure the source and destination databases. The following table describes the parameters.

Section	Parameter	Description
Source Database	Instance Type	Select **RDS Instance** as the type of the source instance.
	Instance Region	The region where the source instance resides.
	RDS Instance ID	Select the ID of the source database.
	Database Account	Enter an account that has the read and write permissions on the source database.
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select **User-Created Database with Public IP Address** as the type of the destination database.
	Instance Region	The region where the destination instance resides.
	Database Type	Select **Oracle**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the Oracle database.
	Port Number	The default port number is 1521.
	Instance Type	◦ **Non-RAC Instance**: If you select this option, you must specify the **SID**. ◦ **RAC or PDB Instance**: If you select this option, you must specify the **Service Name**.
	Database Account	Enter an account that has the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

5. Click **Test Connectivity** and confirm that the test results for both the source and destination databases are **Passed**.

6. In the lower-right corner of the page, click **Set Whitelist and Next**.
7. Select the migration types based on your needs. Select one or more objects from the Available section and click the › icon to move the objects to the **Selected** section.
   - To ensure service continuity during data migration, select **Full Data Migration** and **Incremental Data Migration**.
   - To perform only full data migration, select **Full Data Migration**.
8. Click **Precheck** and wait until the precheck is complete.

   > **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

9. Click **Next** to start the migration task.

   > **Note** If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.4. Migrate data from an ApsaraDB RDS for MySQL instance to a user-created Kafka cluster

Kafka is a distributed message queue service that features high throughput and high scalability. Kafka is widely used for big data analytics such as log collection, data aggregation, streaming processing, and online and offline analysis. It is important for the big data ecosystem. You can use Data Transmission Service (DTS) to migrate data from an ApsaraDB RDS for MySQL instance or a user-created MySQL database to a user-created Kafka cluster. The data migration feature allows you to extend message processing capabilities. This topic uses an ApsaraDB RDS for MySQL instance as an example.

Prerequisites

- If the source database is a user-created MySQL database, binary logging must be enabled for the database and the following requirements must be met:
  - The value of the `binlog_format` parameter is set to `row`.
  - The value of the `binlog_row_image` parameter is set to `full`.
- A Kafka cluster is created and the Kafka version is 0.10 to 2.0.

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- You can select only tables as the objects to be migrated.

Data format

The data that is migrated to the Kafka cluster is stored in the Avro format. You must parse the migrated data based on the Avro schema. For more information, see DTS Avro schema.

Procedure

1. Create a data migration instance.
2. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
3. Configure a task name.

   DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
4. Configure the source and destination databases.

Section	Parameter	Description
Source Database	Instance Type	Select **RDS Instance**.
	Instance Region	Select the region where the source database resides.
	RDS Instance ID	Select the ID of the source RDS instance.
	Database Account	Enter the database account of the source RDS instance. The account must have the SELECT permission on the objects to be migrated, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select **Kafka**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the Kafka cluster. In this example, enter the public IP address.
	Port Number	Enter the service port number of the Kafka cluster. The default port number is 9092.
	Database Account	Enter the username that is used to log on to the Kafka cluster. If no authentication is enabled for the Kafka cluster, you do not need to enter the username.
	Database Password	Enter the password of the username. If no authentication is enabled for the Kafka cluster, you do not need to enter the password.
	Topic	Click **Get Topic List**, and select a topic name from the drop-down list.
	Kafka Version	Select the version of the destination Kafka cluster. **Note** You cannot select Kafka 2.0 in the DTS console. If you are using Kafka 2.0, you must select Kafka 1.0.

Section	Parameter	Description
	Encryption	Select **Non-encrypted** or **SCRAM-SHA-256** based on your business and security requirements.

5. In the lower-right corner of the page, click **Set Whitelist and Next**.
6. Select the migration types and the objects to be migrated.

Setting	Description
Select the migration types	○ To perform only full data migration, select **Schema Migration** and **Full Data Migration**. ○ To ensure service continuity during data migration, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.
Select the objects to be migrated	Select objects from the **Available** section and click the icon to move the objects to the **Selected** section. **Note** ○ You can select columns, tables, or databases as the objects to be migrated. ○ By default, after an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. ○ If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.
Specify the retry time for failed connection to the source or destination database	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 12 hours. You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data migration task. Otherwise, the data migration task fails. **Note** When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

7. In the lower-right corner of the page, click **Precheck**.

Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

8. After the task passes the precheck, click **Next**.

> **Note** If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.5. Migrate data from a user-created SQL Server database to an ApsaraDB RDS for SQL Server instance

This topic describes how to migrate incremental data from a user-created SQL Server database to an ApsaraDB RDS for SQL Server instance by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. You can select all of the supported migration types to ensure service continuity.

Prerequisites

- The version of the user-created SQL Server database is 2005, 2008, 2008 R2, 2012, 2014, 2016, or 2017.

  > **Note**
  > - DTS does not support SQL Server clusters or SQL Server Always On availability groups (AOAGs).
  > - If the version of the source database is 2005, incremental data migration is not supported.
  > - If you migrate data between different versions of databases, make sure that the database versions are compatible.

- The tables to be migrated from the user-created SQL Server database have primary keys or UNIQUE NOT NULL indexes.

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- To ensure that the incremental data migration task runs as expected, do not frequently back up the source database. We recommend that you retain log files for more than three days. Otherwise, you cannot retrieve log files after they are truncated.
- To ensure that the delay time of incremental data migration is accurate, DTS adds a heartbeat table to the user-created SQL Server database. The name of the heartbeat table is `Source table name_dts_mysql_heartbeat`.
- DTS automatically creates a destination database in the ApsaraDB RDS for SQL Server instance. However, if the name of the source database is invalid, you must manually create a database in the ApsaraDB RDS for SQL Server instance before you configure the data migration task.
- If a data migration task fails, DTS automatically resumes the task. Before you switch your workloads to the destination instance, stop or release the data migration task. Otherwise, the data in the source database will overwrite the data in the destination instance after the task is resumed.
- If you perform a primary/secondary switchover on the source database when the data migration task is running, the task fails.
- Migration latency is the difference between the timestamp of the latest migrated data in the destination database and the current timestamp in the source database. If no data manipulation language (DML) operations are performed on the source database for a long time, the migration latency displayed in the DTS console may be inaccurate. If the latency of the migration task is too high, you can perform a DML operation on the source database to update the latency.

> **Note** If you select an entire database as the object to be migrated, you can create a heartbeat table. The heartbeat table is updated or receives data every second.

Limits

- A single data migration task can migrate incremental data from only one database. To migrate incremental data from multiple databases, you must create a data migration task for each database.
- DTS cannot migrate the schemas of assemblies, service brokers, full-text indexes, full-text catalogs, distributed schemas, distributed functions, CLR stored procedures, CLR scalar-valued functions, CLR table-valued functions, internal tables, systems, or aggregate functions.
- DTS cannot migrate data of the sql_variant type.
- DTS cannot migrate tables that contain computed columns.

Migration types

- Schema migration

  DTS migrates the schemas of the required objects to the destination instance. DTS supports schema migration for the following types of objects: table, view, trigger, synonym, SQL stored procedure, SQL function, plan guide, user-defined type, rule, and default.

- Full data migration

  DTS migrates historical data of the required objects from the user-created SQL Server database to the destination database.

- Incremental data migration

  After full data migration is complete, DTS migrates incremental data from the user-created SQL Server database to the destination database. Incremental data migration allows you to ensure service continuity when you migrate data from a user-created SQL Server database to Alibaba Cloud.

SQL operations that can be synchronized during incremental data migration

- INSERT, UPDATE, and DELETE

  > **Note** DTS does not synchronize the UPDATE operations that update only the large fields.

- CREATE TABLE

  > **Note** If a CREATE TABLE operation creates a partition table or a table that contains functions, DTS does not synchronize the operation.

- ALTER TABLE, including only ADD COLUMN, DROP COLUMN, and RENAME COLUMN
- DROP TABLE
- RENAME TABLE, TRUNCATE TABLE, and CREATE INDEX

Procedure

To prevent data migration failures caused by dependencies among objects, DTS migrates the schemas and data from the source SQL Server database in the following order:

1. Migrate the schemas of tables, views, synonyms, user-defined types, rules, defaults, and plan guides.
2. Perform full data migration.
3. Migrate the schemas of SQL stored procedures, SQL functions, triggers, and foreign keys.
4. Perform incremental data migration.

> **Note** During schema migration and full data migration, do not perform data definition language (DDL) operations on the required objects. Otherwise, the objects may fail to be migrated.

Before you begin

Before you configure a data migration task, configure log settings on the user-created SQL Server database.

> **Note** Skip this step if you do not need to perform incremental data migration.

Procedure

1. Create a data migration instance.
2. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
3. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.
	Database Type	Select **SQL Server**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created SQL Server database. In this example, enter the public IP address.
	Port Number	Enter the service port number of the user-created SQL Server database. The default port number is **1433**.
	Database Account	Enter the account that is used to log on to the user-created SQL Server database. **Note** To perform incremental data migration, the account must have the sysadmin permission. To perform schema migration or full data migration, the account must have the SELECT permission on the objects that you want to migrate.
	Database Password	Enter the password of the source database account.
	Instance Type	Select **RDS Instance**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.

Section	Parameter	Description
Destination Database	RDS Instance ID	Select the ID of the destination ApsaraDB RDS for SQL Server instance. **Note** In this scenario, the destination database can be only a SQL Server database.
	Database Account	Enter the database account of the destination ApsaraDB RDS for SQL Server instance. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. If a whitelist is configured for the user-created SQL Server database, you must manually add the CIDR blocks of DTS servers to the whitelist of the database. To obtain the CIDR blocks of DTS servers, click **Copy to Clipboard** in the dialog box that appears. After you add the CIDR blocks of DTS servers to the whitelist of the database, click **Next**.

   Note If you do not need to configure a whitelist for the user-created SQL Server database, ignore the preceding settings and click **Next**.

6. Select the migration types and the objects to be migrated.

Setting	Description
Select the migration types	○ To perform only full data migration, select **Schema Migration** and **Full Data Migration**. ○ To ensure service continuity during data migration, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.
Select the objects to be migrated	Select objects from the **Available** section and click the › icon to move the objects to the **Selected** section. **Note** ○ You can select columns, tables, or databases as the objects to be migrated. ○ By default, after an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. ○ If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.

Setting	Description
Specify the retry time for failed connection to the source or destination database	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 12 hours. You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data migration task. Otherwise, the data migration task fails. **Note** When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

7. In the lower-right corner of the page, click **Precheck**.

   Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

8. After the task passes the precheck, click **Next**.

   Note If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.6. Migrate data between user-created Oracle databases

This topic describes how to migrate data between user-created Oracle databases by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. You can select all of the supported migration types to ensure service continuity.

Prerequisites

- The version of the source Oracle database is 9i, 10g, 11g, 12c, 18c, or 19c.

  Note To ensure compatibility, make sure the versions of the source and destination databases are the same.

- Supplemental logging, including SUPPLEMENTAL_LOG_DATA_PK and SUPPLEMENTAL_LOG_DATA_UI, is enabled for the source Oracle database. For more information, see Supplemental Logging.
- The source Oracle database is running in ARCHIVELOG mode. Archived log files are accessible and a suitable retention period is set for archived log files. For more information, see Managing Archived Redo Log Files.
- The available storage space of the destination Oracle database is larger than the total size of the data in the source Oracle database.
- If you perform a primary/secondary switchover on the source database when the data migration task is running, the task fails.

Migration types

Migration type	Description
Schema migration	DTS migrates the schemas of the required objects to the destination Oracle database. DTS supports schema migration for the following types of objects: table, view, synonym, trigger, stored procedure, function, package, and user-defined type. **Note** If an object contains triggers, the data between the source and destination databases will become inconsistent.
Full data migration	DTS migrates historical data of the required objects from the source Oracle database to the destination Oracle database. **Note** During schema migration and full data migration, do not perform DDL operations on the objects to be migrated. Otherwise, the objects may fail to be migrated.
Incremental data migration	After full data migration, DTS retrieves redo log files from the source Oracle database. Then, DTS synchronizes incremental data from the source Oracle database to the destination Oracle database. Incremental data migration allows you to ensure service continuity when you migrate data between Oracle databases.

Permissions required for database accounts

Database	Schema migration	Full data migration	Incremental data migration
Source Oracle database	The owner permission on schemas	The owner permission on schemas	SYSDBA
Destination Oracle database	The owner permission on schemas	The owner permission on schemas	The owner permission on schemas

Note For more information about how to create and authorize an Oracle database account, see CREATE USER and GRANT.

Procedure

1. Create a data migration instance.
2. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
3. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.

Section	Parameter	Description
Source Database	Database Type	Select **Oracle**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the source Oracle database.
	Port Number	Enter the port number of the source Oracle database. The default port number is 1521.
	Instance Type	Select **Non-RAC Instance** or **RAC or PDB Instance** based on the architecture of the source Oracle database.
	SID	Enter the system ID (SID) of the source Oracle database. **Note** This parameter is required if you select **Non-RAC Instance** as the instance type.
	Service Name	Enter the server name of the instance. **Note** This parameter is required if you select **RAC or PDB Instance** as the instance type.
	Database Account	Enter the account that is used to connect to the source Oracle database.
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select **Oracle**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the destination Oracle database.
	Port Number	Enter the port number of the destination Oracle database. The default port number is 1521.
	Instance Type	Select **Non-RAC Instance** or **RAC or PDB Instance** based on the architecture of the destination Oracle database.
	SID	Enter the SID of the destination Oracle database. **Note** This parameter is required if you select **Non-RAC Instance** as the instance type.

Section	Parameter	Description
	Service Name	Enter the server name of the instance. **Note** This parameter is required if you select **RAC or PDB Instance** as the instance type.
	Database Account	Enter the account that is used to connect to the destination Oracle database.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. If your Oracle databases have security settings, you must manually add the CIDR blocks of DTS servers to the whitelists of the databases. To obtain the CIDR blocks of DTS servers, click **Copy to Clipboard** in the dialog box that appears. After you add the CIDR blocks of DTS servers to the whitelist of the databases, click **Next**.

   Note If you do not need to configure whitelists for your Oracle databases, ignore the preceding settings and click **Next**.

6. Select the migration types and the objects to be migrated.

Setting	Description
Select the migration types	○ To perform only full data migration, select **Schema Migration** and **Full Data Migration**. ○ To ensure service continuity during data migration, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.
Select the objects to be migrated	Select objects from the **Available** section and click the › icon to move the objects to the **Selected** section. **Note** ○ You can select columns, tables, or databases as the objects to be migrated. ○ By default, after an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. ○ If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.

Setting	Description
Specify the retry time for failed connection to the source or destination database	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 12 hours. You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data migration task. Otherwise, the data migration task fails. **Note** When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

7. In the lower-right corner of the page, click **Precheck**.

   Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

8. After the task passes the precheck, click **Next**.

   Note If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.7. Migrate data from a user-created Oracle database to an ApsaraDB RDS for MySQL instance

This topic describes how to migrate data from a user-created Oracle database to an ApsaraDB RDS for MySQL instance by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. When you migrate data from a user-created Oracle database, you can select all of the supported migration types to ensure service continuity.

Prerequisites

- The version of the user-created Oracle database is 9i, 10g, 11g, 12c, 18c, or 19c.
- Supplemental logging, including SUPPLEMENTAL_LOG_DATA_PK and SUPPLEMENTAL_LOG_DATA_UI, is enabled for the user-created Oracle database. For more information, see Supplemental Logging.
- The user-created Oracle database is running in ARCHIVELOG mode. Archived log files are accessible and a suitable retention period is set for archived log files. For more information, see Managing Archived Redo Log Files.

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- Table names in the ApsaraDB RDS for MySQL instance are case-insensitive. If a table name in the user-created Oracle database contains uppercase letters, ApsaraDB RDS for MySQL converts all uppercase letters to

lowercase letters before creating the table.

If the source Oracle database contains identical table names that differ only in capitalization, these table names are identified as duplicate. During schema migration, the following message is returned: "The object already exists". To prevent name conflicts in the destination database, you can change the names of the migrated objects by using the object name mapping feature. For more information, see Object name mapping.

- If you perform a primary/secondary switchover on the source database when the data migration task is running, the task fails.

Migration types

- Schema migration

  DTS supports schema migration for tables and indexes. DTS does not support schema migration for the following types of objects: view, synonym, trigger, stored procedure, function, package, and user-defined type. DTS has the following limits on schema migration for tables and indexes:

  - Schema migration of nested tables is not supported. Clustered tables and index-organized tables (IOTs) are converted into common tables in the destination database.
  - Schema migration of function-based indexes, domain indexes, bitmap indexes, and reverse indexes is not supported.

- Full data migration

  DTS migrates historical data of the required objects from the user-created Oracle database to the destination ApsaraDB RDS for MySQL instance.

- Incremental data migration

  DTS retrieves redo log files from the user-created Oracle database. Then, DTS synchronizes incremental data from the user-created Oracle database to the destination database in the ApsaraDB RDS for MySQL instance. Incremental data migration allows you to ensure service continuity when you migrate data from the user-created Oracle database to the destination database.

SQL operations that can be synchronized during incremental data migration

- INSERT, DELETE, and UPDATE
- CREATE TABLE

  > **Note** If a CREATE TABLE operation creates a table that contains functions, DTS does not synchronize the operation.

- ALTER TABLE, ADD COLUMN, DROP COLUMN, RENAME COLUMN, and ADD INDEX
- DROP TABLE
- RENAME TABLE, TRUNCATE TABLE, and CREATE INDEX

Data type mappings

For more information, see Data type mappings between heterogeneous databases.

Procedure

1. Create a data migration instance.
2. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
3. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select an instance type based on the deployment of the source database. In this example, select **User-Created Database with Public IP Address**.
	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.
	Database Type	Select **Oracle**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created Oracle database.
	Port Number	Enter the service port number of the user-created Oracle database. The default port number is **1521**.
	Instance Type	○ **Non-RAC Instance**: If you select this option, you must specify the **SID**. ○ **RAC or PDB Instance**: If you select this option, you must specify the **Service Name**.
	Database Account	Enter the account of the user-created Oracle database. To perform incremental data migration, the account must have the database administrator (DBA) permission. To perform schema migration or full data migration, the account must have the owner permission on schemas.
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select **RDS Instance**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	RDS Instance ID	Select the ID of the destination ApsaraDB RDS for MySQL instance.
	Database Account	Enter the database account of the destination ApsaraDB RDS for MySQL instance. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. If your Oracle databases have security settings, you must manually add the CIDR blocks of DTS servers to the whitelists of the databases. To obtain the CIDR blocks of DTS servers, click **Copy to Clipboard** in the dialog box that appears. After you add the CIDR blocks of DTS servers to the whitelist of the databases, click **Next**.

   Note If you do not need to configure whitelists for your Oracle databases, ignore the preceding settings and click **Next**.

6. Select the migration types and the objects to be migrated.

Setting	Description
Select the migration types	◦ To perform only full data migration, select **Schema Migration** and **Full Data Migration**. ◦ To ensure service continuity during data migration, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.
Select the objects to be migrated	Select objects from the **Available** section and click the icon to move the objects to the **Selected** section. **Note** ◦ You can select columns, tables, or databases as the objects to be migrated. ◦ By default, after an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. ◦ If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.
Specify the retry time for failed connection to the source or destination database	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 12 hours. You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data migration task. Otherwise, the data migration task fails. **Note** When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

7. In the lower-right corner of the page, click **Precheck**.

   Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

8. After the task passes the precheck, click **Next**.

   Note If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.8. Migrate data from a user-created Oracle database to a PolarDB cluster

This topic describes how to migrate data from a user-created Oracle database to a PolarDB cluster by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. You can select all of the supported migration types to ensure service continuity.

Prerequisites

- The version of the user-created Oracle database is 9i, 10g, 11g, 12c, 18c, or 19c.
- Supplemental logging, including SUPPLEMENTAL_LOG_DATA_PK and SUPPLEMENTAL_LOG_DATA_UI, is enabled for the user-created Oracle database. For more information, see Supplemental Logging.
- The user-created Oracle database is running in ARCHIVELOG mode. Archived log files are accessible and a suitable retention period is set for archived log files. For more information, see Managing Archived Redo Log Files.

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- DTS supports schema migration for the following types of objects: table, view, synonym, trigger, stored procedure, function, package, and user-defined type.
- If you perform a primary/secondary switchover on the source database when the data migration task is running, the task fails.

Limits

- During schema migration, the reverse indexes and bitmap indexes of the source database are stored as common indexes in the PolarDB cluster.
- During schema migration, partitioned indexes are converted into independent indexes on each partition in the PolarDB cluster.
- Incremental data migration supports only tables that have primary keys or UNIQUE NOT NULL indexes.
- Incremental data migration does not support the LONG data type.
- Data definition language (DDL) operations that are performed during incremental data migration cannot be synchronized to the destination database.
- Materialized views cannot be migrated.

Data type mappings

For more information, see Data type mappings between heterogeneous databases.

Procedure

1. Create a data migration instance.
2. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
3. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.

Section	Parameter	Description
Source Database	Instance Type	Select an instance type based on the deployment of the source database. In this example, select **User-Created Database with Public IP Address**.
	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.
	Database Type	Select **Oracle**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created Oracle database.
	Port Number	Enter the service port number of the user-created Oracle database. The default port number is **1521**.
	Instance Type	○ **Non-RAC Instance**: If you select this option, you must specify the **SID**. ○ **RAC or PDB Instance**: If you select this option, you must specify the **Service Name**.
	Database Account	Enter the account of the user-created Oracle database. To perform incremental data migration, the account must have the database administrator (DBA) permission. To perform schema migration or full data migration, the account must have the owner permission on schemas.
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select **PolarDB-O**.
	DNS or IP Address	The endpoint of the destination PolarDB cluster.
	Port Number	Enter the service port number of the destination PolarDB cluster. The default port number is **1433**.
	Database Name	Enter the name of the destination database in the PolarDB cluster.
	Database Account	Enter the database account of the destination PolarDB cluster. The account must have the read and write permissions.
	Database Password	Enter the password of the destination database account. **Note** After you specify the information about the RDS instance, you can click **Test Connectivity** next to **Database Password** to check whether the information is valid. If the information is valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Then, modify the information based on the check results.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. If your Oracle databases have security settings, you must manually add the CIDR blocks of DTS servers to the whitelists of the databases. To obtain the CIDR blocks of DTS servers, click **Copy to Clipboard** in the dialog

box that appears. After you add the CIDR blocks of DTS servers to the whitelist of the databases, click **Next**.

> **Note** If you do not need to configure whitelists for your Oracle databases, ignore the preceding settings and click **Next**.

6. Select the migration types and the objects to be migrated.

Setting	Description
Select the migration types	○ To perform only full data migration, select **Schema Migration** and **Full Data Migration**. ○ To ensure service continuity during data migration, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.
Select the objects to be migrated	Select objects from the **Available** section and click the › icon to move the objects to the **Selected** section. **Note** ○ You can select columns, tables, or databases as the objects to be migrated. ○ By default, after an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. ○ If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.
Specify the retry time for failed connection to the source or destination database	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 12 hours. You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data migration task. Otherwise, the data migration task fails. **Note** When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

7. In the lower-right corner of the page, click **Precheck**.

> **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

8. After the task passes the precheck, click **Next**.

> **Note** If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.9. Migrate data from a user-created Oracle database to a PolarDB-X instance

This topic describes how to migrate data from a user-created Oracle database to a PolarDB-X instance by using Data Transmission Service (DTS). PolarDB-X is formerly known as Distributed Relational Database Service (DRDS). DTS allows you to ensure service continuity when you migrate both historical data and incremental data.

Prerequisites

- The version of the user-created Oracle database is 9i, 10g, 11g, 12c, 18c, or 19c.
- Supplemental logging, including SUPPLEMENTAL_LOG_DATA_PK and SUPPLEMENTAL_LOG_DATA_UI, is enabled for the user-created Oracle database. For more information, see Supplemental Logging.
- The user-created Oracle database is running in ARCHIVELOG mode. Archived log files are accessible and a suitable retention period is set for archived log files. For more information, see Managing Archived Redo Log Files.

Precautions

- DTS cannot migrate schemas from a user-created Oracle database to a PolarDB-X instance.

  > **Note** During schema migration, DTS migrates the schemas of the required objects, such as tables, from the source database to the destination database.

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- If you perform a primary/secondary switchover on the source database when the data migration task is running, the task fails.

Migration types

- Full data migration

  DTS migrates historical data of the required objects from the source Oracle database to the destination database.

- Incremental data migration

  After full data migration is complete, DTS retrieves redo log files from the source Oracle database. Then, DTS synchronizes incremental data from the source Oracle database to the destination database.

  > **Note** The following SQL operations can be synchronized during incremental data migration: INSERT, DELETE, and UPDATE. Data definition language (DDL) operations cannot be synchronized during incremental data migration.

Procedure

1. Create databases and tables in the destination PolarDB-X instance based on the schemas of the source

tables.

> **Note** The data types of Oracle and PolarDB-X do not have one-to-one correspondence. You must create the corresponding data types in the destination PolarDB-X instance. For more information, see Data type mappings between heterogeneous databases.

2. Create a data migration instance.
3. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
4. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.
	Database Type	Select **Oracle**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created Oracle database. In this example, enter the public IP address.
	Port Number	Enter the service port number of the user-created Oracle database. The default port number is **1521**.
	Instance Type	○ **Non-RAC Instance**: If you select this option, you must specify the **SID**. ○ **RAC or PDB Instance**: If you select this option, you must specify the **Service Name**.
	Database Account	Enter the account of the user-created Oracle database. To perform incremental data migration, the account must have the database administrator (DBA) permission. To perform schema migration or full data migration, the account must have the owner permission on schemas.
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select **DRDS Instance**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the destination PolarDB-X instance.
	Database Account	Enter the database account of the destination PolarDB-X instance. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

5. In the lower-right corner of the page, click **Set Whitelist and Next**.

6. If your Oracle databases have security settings, you must manually add the CIDR blocks of DTS servers to the whitelists of the databases. To obtain the CIDR blocks of DTS servers, click **Copy to Clipboard** in the dialog box that appears. After you add the CIDR blocks of DTS servers to the whitelist of the databases, click **Next**.

   > **Note** If you do not need to configure whitelists for your Oracle databases, ignore the preceding settings and click **Next**.

7. Select the migration types and the objects to be migrated.

Setting	Description
Select the migration types	◦ To perform only full data migration, select only **Full Data Migration**. ◦ To ensure service continuity during data migration, select both **Full Data Migration** and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.
Select the objects to be migrated	Select one or more objects from the **Available** section and click the › icon to move the objects to the **Selected** section. **Note** ◦ You can select columns, tables, or databases as the objects to be migrated. ◦ By default, after an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. ◦ If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.
Specify the retry time for failed connection to the source or destination database	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 12 hours. You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data migration task. Otherwise, the data migration task fails. **Note** When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

8. In the lower-right corner of the page, click **Precheck**.

   > **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

9. After the task passes the precheck, click **Next**.

> **Note** If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.10. Migrate data between ApsaraDB RDS for PostgreSQL instances

This topic describes how to migrate data between ApsaraDB RDS for PostgreSQL instances by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. You can select all of the supported migration types to ensure service continuity.

Prerequisites

The network type of both the source and destination ApsaraDB RDS for PostgreSQL instances is VPC.

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- To ensure that the delay time of incremental data migration is accurate, DTS adds a heartbeat table named *dts_postgres_heartbeat* to the source database.
- During incremental data migration, DTS creates a replication slot for the source database. The replication slot is prefixed with `dts_sync_`. DTS automatically clears historical replication slots every 90 minutes to reduce storage usage.

  > **Note** If the migration task is released or fails, DTS automatically clears the replication slot. If a primary/secondary switchover is performed on the source ApsaraDB RDS for PostgreSQL instance, you must log on to the secondary database to clear the replication slot.

- If a data migration task fails, DTS automatically resumes the task. Before you switch your workloads to the destination instance, stop or release the data migration task. Otherwise, the data in the source instance will overwrite the data in the destination instance after the task is resumed.

Limits

- A single data migration task can migrate data from only one database. To migrate data from multiple databases, you must create a data migration task for each database.
- During incremental data migration, DTS migrates only data manipulation language (DML) operations. DML operations include INSERT, DELETE, and UPDATE.

Data migration process

The following table describes how DTS migrates the schemas and data of the source PostgreSQL database. The process prevents data migration failures that are caused by dependencies between objects.

Data migration process	Description
1. Schema migration	DTS migrates the schemas of tables, views, sequences, functions, user-defined types, rules, domains, operations, and aggregates to the destination database. **Note** DTS does not migrate functions that are written in the C programming language.
2. Full data migration	DTS migrates historical data of the required objects to the destination database.
3. Schema migration	DTS migrates the schemas of triggers and foreign keys to the destination database.
4. Incremental data migration	DTS migrates incremental data of the required objects to the destination database. **Note** Incremental data migration does not support the BIT data type.

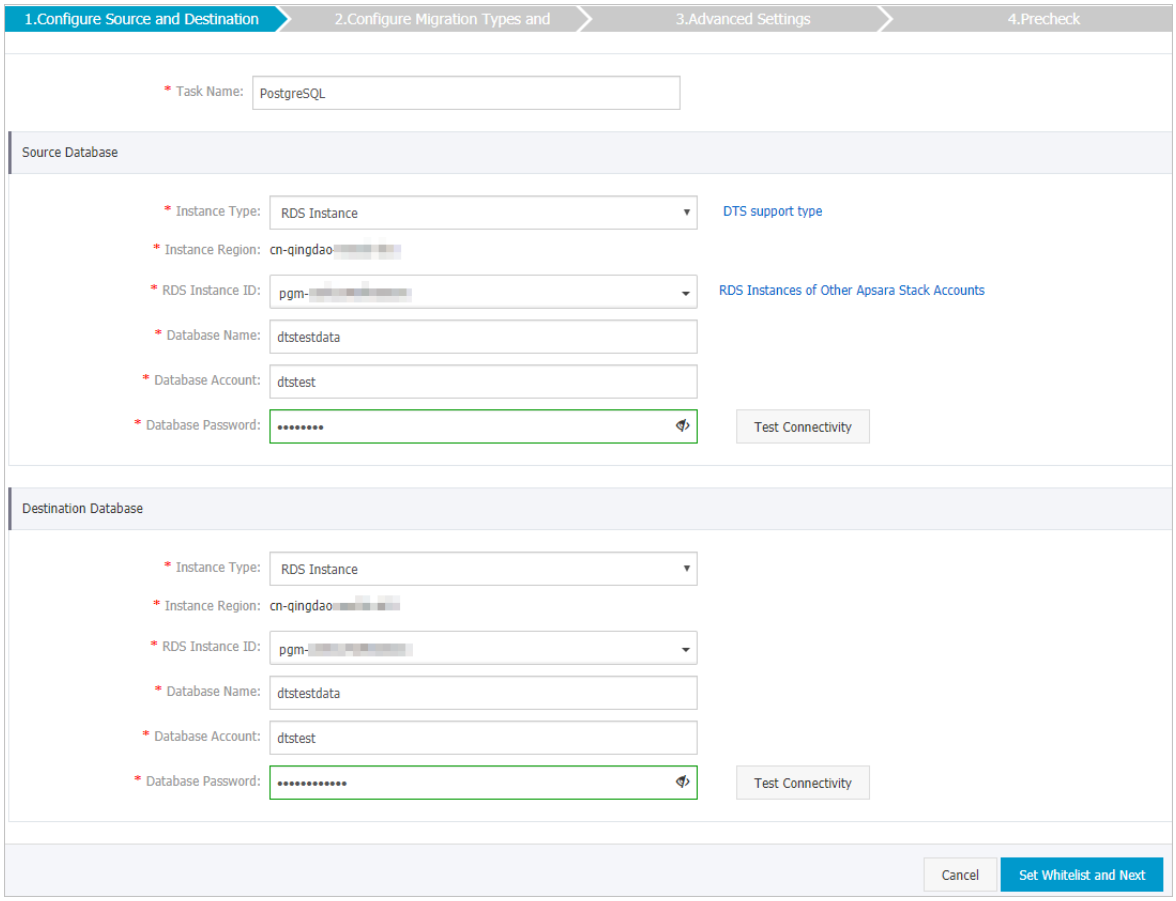
Before you begin

If you need to perform incremental data migration, you must perform the following steps: Log on to the ApsaraDB RDS console. Click the ID of the source ApsaraDB RDS for PostgreSQL instance. In the left-side navigation pane, click **Parameters**. Find the wal_level parameter and change the value to logical.

Warning After you change the value of the `wal_level` parameter, you must restart the instance to apply the change. We recommend that you evaluate the impact on your business and change the parameter setting during off-peak hours.

Procedure

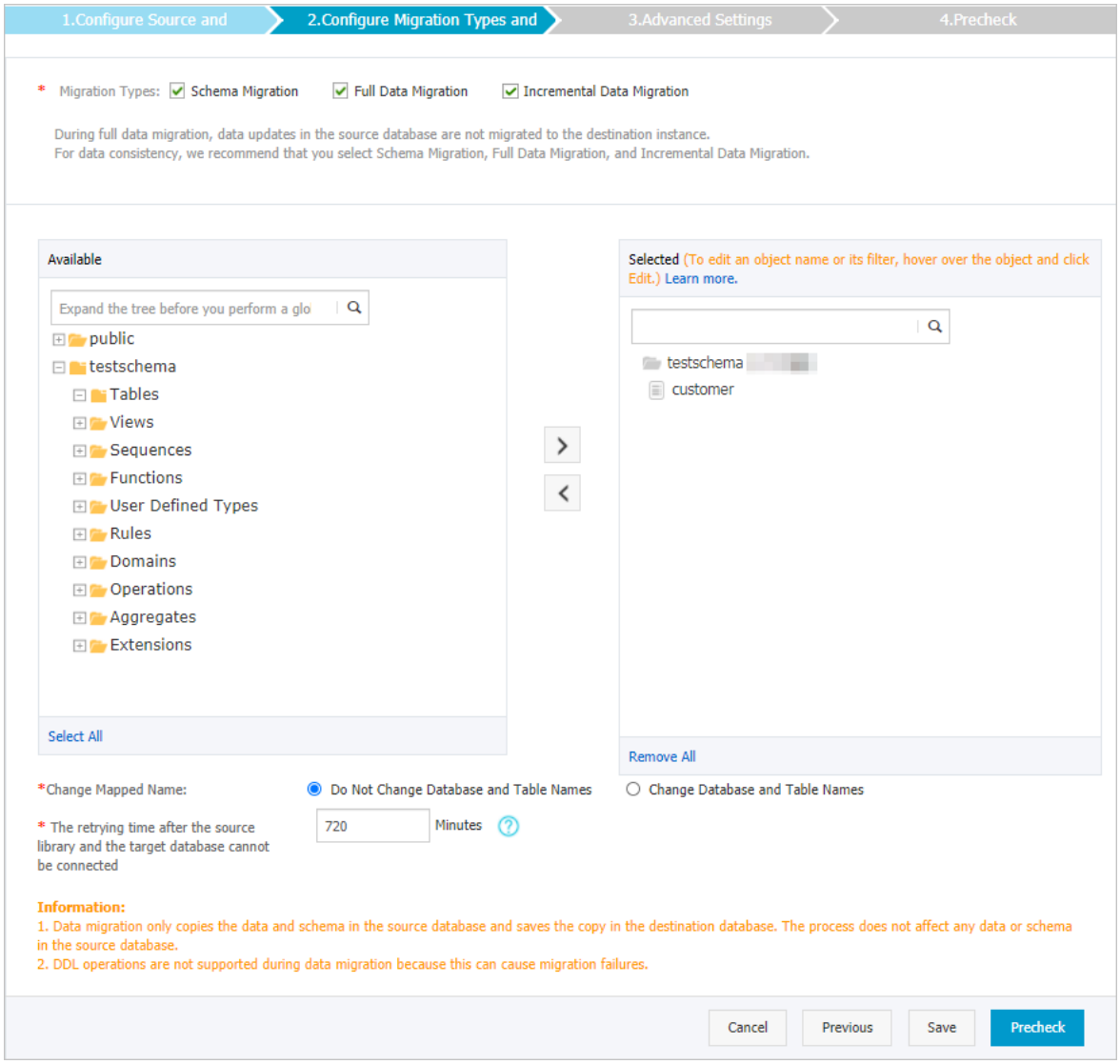
1. Create a data migration instance.
2. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
3. Configure the source and destination databases.



Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select **RDS Instance**.
	Instance Region	Select the region where the source ApsaraDB RDS for PostgreSQL instance resides.
	RDS Instance ID	Select the ID of the source ApsaraDB RDS for PostgreSQL instance.
	Database Name	Enter the name of the source database.
	Database Account	Enter the database account of the source ApsaraDB RDS for PostgreSQL instance. The account must have the read and write permissions on the source database.
	Database Password	Enter the password of the source database account.
	Instance Type	Select **RDS Instance**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	RDS Instance ID	Select the ID of the destination ApsaraDB RDS for PostgreSQL instance.

Section	Parameter	Description
Destination Database	Database Name	Enter the name of the destination database. **Note** The name of the destination database can be different from the name of the source database.
	Database Account	Enter the database account of the destination ApsaraDB RDS for PostgreSQL instance. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Select the migration types and the objects to be migrated.



Setting	Description
Select the migration types	◦ To perform only full data migration, select **Schema Migration** and **Full Data Migration**. ◦ To ensure service continuity during data migration, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.
Select the objects to be migrated	Select objects from the **Available** section and click the icon to move the objects to the **Selected** section. **Note** ◦ You can select columns, tables, or databases as the objects to be migrated. ◦ By default, after an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. ◦ If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.
Specify the retry time for failed connection to the source or destination database	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 12 hours. You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data migration task. Otherwise, the data migration task fails. **Note** When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

6. In the lower-right corner of the page, click **Precheck**.

   Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. After the task passes the precheck, click **Next**.

   Note If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.11. Migrate data between PolarDB clusters

This topic describes how to migrate data between PolarDB clusters by using Data Transmission Service (DTS).

Prerequisites

.

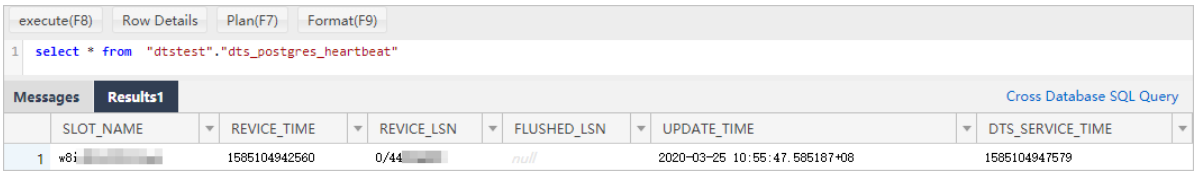
The tables to be migrated from the source PolarDB cluster contain primary keys or UNIQUE NOT NULL indexes.

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- A single data migration task can migrate data from only one database. To migrate data from multiple databases, you must create a data migration task for each database.
- If you select a schema as the object to be migrated and create a table in the schema or run the RENAME command to rename the table, you must run the `ALTER TABLE schema.table REPLICA IDENTITY FULL;` command before you write data to the table.

  > **Note** Replace the `schema` and `table` in the preceding sample command with the actual schema name and table name.

- To ensure that the delay time of data migration is accurate, DTS adds a heartbeat table named `dts_postgres_heartbeat` to the source database. The following figure shows the schema of the heartbeat table.



- If a data migration task fails, DTS automatically resumes the task. Before you switch your workloads to the destination database, stop or release the data migration task. Otherwise, the data from the source database will overwrite the data in the destination database after the task is resumed.

Migration types

Migration type	Description
Schema migration	DTS migrates the schemas of the required objects from the source database to the destination PolarDB cluster. DTS supports schema migration for the following types of objects: table, view, synonym, trigger, stored procedure, function, package, and user-defined type. **Notice** However, if an object contains triggers, data will become inconsistent between the source and destination databases.
Full data migration	DTS migrates historical data of the required objects from the source database to the destination PolarDB cluster. **Notice** During schema migration and full data migration, do not perform data definition language (DDL) operations on the required objects. Otherwise, the objects may fail to be migrated.

Migration type	Description
Incremental data migration	DTS retrieves redo log files from the source database. Then, DTS synchronizes incremental data from the source database to the destination PolarDB cluster. DTS can synchronize data manipulation language (DML) operations, including INSERT, UPDATE, and DELETE operations. DTS cannot synchronize DDL operations. Incremental data migration allows you to ensure service continuity when you migrate data between PolarDB clusters.

Procedure

1. Create a data migration instance.
2. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
3. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select **User-Created Database with Public IP Address**. You cannot select PolarDB cluster as the instance type.
	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.
	Database Type	Select **PolarDB-O**.
	DNS or IP Address	The endpoint of the source PolarDB cluster.
	Port Number	Enter the service port number of the PolarDB cluster. The default port number is **3433**.
	Database Name	Enter the name of the source database.
	Database Account	Enter the initial account of the source PolarDB cluster.
	Database Password	Enter the password of the source database account. **Note** After you specify the information about the self-managed Oracle database, you can click **Test Connectivity** next to **Database Password** to check whether the information is valid. If the information is valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Then, modify the information based on the check results.
	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select **PolarDB-O**.

Section	Parameter	Description
Destination Database	DNS or IP Address	The endpoint of the destination PolarDB cluster.
	Port Number	Enter the service port number of the destination PolarDB cluster. The default port number is **1433**.
	Database Name	Enter the name of the destination database.
	Database Account	Enter the database account of the destination PolarDB cluster. The account must have the read and write permissions.
	Database Password	Enter the password of the destination database account. **Note** After you specify the information about the RDS instance, you can click **Test Connectivity** next to **Database Password** to check whether the information is valid. If the information is valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Then, modify the information based on the check results.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Select the migration types and the objects to be migrated.

Setting	Description
Select the migration types	◦ To perform only full data migration, select **Schema Migration** and **Full Data Migration**. ◦ To ensure service continuity during data migration, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.

Setting	Description
Select the objects to be migrated	Select objects from the **Available** section and click the icon to move the objects to the **Selected** section. **Note** • You can select columns, tables, or databases as the objects to be migrated. • By default, after an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. • If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.
Specify the retry time for failed connection to the source or destination database	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 12 hours. You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data migration task. Otherwise, the data migration task fails. **Note** When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

6. In the lower-right corner of the page, click **Precheck**.

   Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. After the task passes the precheck, click **Next**.

   Note If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.12. Migrate data from a PolarDB cluster to a user-created Oracle database

This topic describes how to migrate data from a PolarDB cluster to a user-created Oracle database by using Data Transmission Service (DTS).

Prerequisites

The destination Oracle database is created. The schema of the Oracle database is the same as the schema of the source database in the PolarDB cluster. This is because DTS does not support schema migration from a PolarDB cluster to a user-created Oracle database.

Permissions required for database accounts

Migration types	Full data migration	Incremental data migration
Source PolarDB cluster	The read and write permissions	The read and write permissions
Destination Oracle database	The read and write permissions	The read and write permissions

Procedure

1. Create a data migration instance.
2. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
3. Configure the source and destination instances.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select **User-Created Database with Public IP Address**. You cannot select PolarDB cluster as the instance type.
	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.
	Database Type	Select **PolarDB-O**.
	DNS or IP Address	Enter the endpoint of the source PolarDB cluster.
	Port Number	Enter the service port number of the source PolarDB cluster. The default port number is **1521**.
	Database Name	Enter the name of the source database.
	Database Account	Enter an account that has the read and write permissions on the source database.

Section	Parameter	Description
	Database Password	Enter the password of the source database account. **Note** After you specify the information about the self-managed Oracle database, you can click **Test Connectivity** next to **Database Password** to check whether the information is valid. If the information is valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Then, modify the information based on the check results.
Destination Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select **Oracle**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created Oracle database.
	Port Number	Enter the service port number of the user-created Oracle database. The default port number is **1521**.
	Instance Type	○ **Non-RAC Instance**: If you select this option, you must specify the **SID**. ○ **RAC or PDB Instance**: If you select this option, you must specify the **Service Name**.
	Database Account	Enter an account that has the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account. **Note** After you specify the information about the RDS instance, you can click **Test Connectivity** next to **Database Password** to check whether the information is valid. If the information is valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Then, modify the information based on the check results.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Select the migration types and the objects to be migrated.

Setting	Description

Setting	Description
Select the migration types	◦ To perform only full data migration, select only **Full Data Migration**. ◦ To ensure service continuity during data migration, select both **Full Data Migration** and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.
Select the objects to be migrated	Select one or more objects from the **Available** section and click the › icon to move the objects to the **Selected** section. **Note** ◦ You can select columns, tables, or databases as the objects to be migrated. ◦ By default, after an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. ◦ If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.
Specify the retry time for failed connection to the source or destination database	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 12 hours. You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data migration task. Otherwise, the data migration task fails. **Note** When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

6. In the lower-right corner of the page, click **Precheck**.

   Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. After the task passes the precheck, click **Next**.

   Note If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.13. Migrate data from a PolarDB cluster to a user-created Kafka cluster

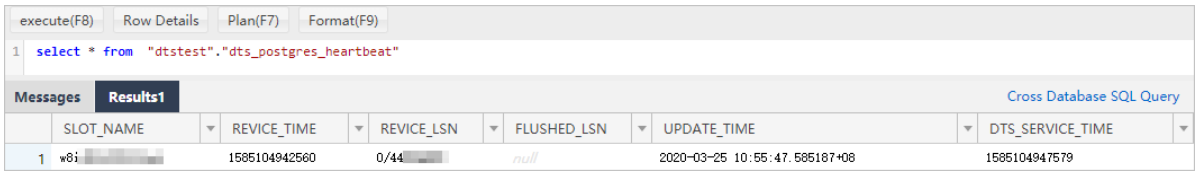
Kafka is a distributed message queue service that features high throughput and high scalability. Kafka is widely used for big data analytics such as log collection, data aggregation, streaming processing, and online and offline analysis. It is important for the big data ecosystem. This topic describes how to migrate data from a PolarDB cluster to a user-created Kafka cluster by using Data Transmission Service (DTS). The data migration feature allows you to extend message processing capabilities.

Prerequisites

The tables to be migrated from the source PolarDB cluster contain primary keys or UNIQUE NOT NULL indexes.

Precautions

- DTS can migrate only incremental data from a PolarDB cluster to a user-created Kafka cluster.
- A single data migration task can migrate data from only one database. To migrate data from multiple databases, you must create a data migration task for each database.
- To ensure that the delay time of data migration is accurate, DTS adds a heartbeat table named `dts_postgres_heartbeat` to the source database. The following figure shows the schema of the heartbeat table.



Procedure

1. Create a data migration instance.
2. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
3. Configure the source and destination instances.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select **User-Created Database with Public IP Address**. You cannot select PolarDB cluster as the instance type.
	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.
	Database Type	Select **PolarDB-O**.
	DNS or IP Address	Enter the endpoint of the source PolarDB cluster.
	Port Number	Enter the service port number of the source PolarDB cluster. The default port number is **1521**.
	Database Name	Enter the name of the source database.
	Database Account	Enter the privileged account of the source PolarDB cluster.

Section	Parameter	Description
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select **Kafka**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the Kafka cluster. In this example, enter the public IP address.
	Port Number	Enter the service port number of the Kafka cluster. The default port number is 9092.
	Database Account	Enter the username that is used to log on to the Kafka cluster. If no authentication is enabled for the Kafka cluster, you do not need to enter the username.
	Database Password	Enter the password of the database account. If no authentication is enabled for the Kafka cluster, you do not need to enter the password.
	Topic	Click **Get Topic List**, and select a topic name from the drop-down list.
	Kafka Version	Select the version of the user-created Kafka cluster. **Note** You cannot select Kafka 2.0 in the DTS console. If you are using Kafka 2.0, you must select Kafka 1.0.
	Encryption	Select **Non-encrypted** or **SCRAM-SHA-256** based on your business and security requirements.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Select the migration type and the objects to be migrated.

Setting	Description
Select the migration type	Select **Incremental Data Migration**.
Select the objects to be migrated	Select one or more tables from the **Available** section and click the icon to move the tables to the **Selected** section. You can select only tables as the objects to be migrated.

6. In the lower-right corner of the page, click **Precheck**.

Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. After the task passes the precheck, click **Next**.

> **Note** If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.14. Migrate data between user-created Redis databases

This topic describes how to migrate data between user-created Redis databases by using Data Transmission Service (DTS).

Prerequisites

- The version of the source Redis database is 2.8, 3.0, 3.2, 4.0, or 5.0.
- The source Redis database uses the standalone architecture rather than the cluster architecture.
- The `PSYNC` or `SYNC` command can be executed on the source Redis database.

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- If the data eviction policy (`maxmemory-policy`) of the destination database is not set to `noeviction` , data may become inconsistent between the source and destination databases. For more information, see Eviction policies.
- If you run the EVAL or EVALSHA command to call Lua scripts, DTS cannot identify whether these Lua scripts are executed on the destination database. During incremental data migration, the destination database does not explicitly return the execution results of Lua scripts.
- When you run the `PSYNC` or `SYNC` command to transmit data of the LIST type, DTS does not perform the `flush` operation on the existing data. Therefore, the destination database may contain duplicate data records.
- Migration latency is the difference between the timestamp of the latest migrated data in the destination database and the current timestamp in the source database. If no data manipulation language (DML) operations are performed on the source database for a long time, the migration latency displayed in the DTS console may be inaccurate. If the latency of the migration task is too high, you can perform a DML operation on the source database to update the latency.

> **Note** If you select an entire database as the object to be migrated, you can create a heartbeat table. The heartbeat table is updated or receives data every second.

Migration types

- Full data migration: DTS migrates historical data of the required objects from the source database to the destination database.
- Incremental data migration: DTS migrates incremental data from the source database to the destination database in real time.

Operations that can be synchronized during incremental data migration

- APPEND
- BITOP, BLPOP, BRPOP, and BRPOPLPUSH
- DECR, DECRBY, and DEL

- EVAL, EVALSHA, EXEC, EXPIRE, and EXPIREAT
- FLUSHALL and FLUSHDB
- GEOADD and GETSET
- HDEL, HINCRBY, HINCRBYFLOAT, HMSET, HSET, and HSETNX
- INCR, INCRBY, and INCRBYFLOAT
- LINSERT, LPOP, LPUSH, LPUSHX, LREM, LSET, and LTRIM
- MOVE, MSET, MSETNX, and MULTI
- PERSIST, PEXPIRE, PEXPIREAT, PFADD, PFMERGE, PSETEX, and PUBLISH
- RENAME, RENAMENX, RESTORE, RPOP, RPOPLPUSH, RPUSH, and RPUSHX
- SADD, SDIFFSTORE, SELECT, SET, SETBIT, SETEX, SETNX, SETRANGE, SINTERSTORE, SMOVE, SPOP, SREM, and SUNIONSTORE
- ZADD, ZINCRBY, ZINTERSTORE, ZREM, ZREMRANGEBYLEX, ZUNIONSTORE, ZREMRANGEBYRANK, and ZREMRANGEBYSCORE

Before you begin

To ensure that incremental data migration tasks run as expected, we recommend that you remove the limit on the replication output buffer for the source database. This topic uses a Linux server as an example.

> **Note** If you perform only full data migration, skip the following steps.

1. Use the redis-cli program to connect to the source database.

   > **Note** You can use the redis-cli program after you install the Redis client. For more information, see Redis community official website.

   ```
   redis-cli -h <host> -p <port> -a <password>
   ```

   > **Note**
   > - <host>: the endpoint that is used to connect to the source database. You can use 127.0.0.1 in this example.
   > - <port>: the service port number of the source database. The default port number is 6379.
   > - <password>: the password of the source database.

   Example:

   ```
   redis-cli -h 127.0.0.1 -p 6379 -a Test123456
   ```

2. Run the following command to remove the limit on the replication output buffer:

   ```
   config set client-output-buffer-limit 'slave 0 0 0'
   ```

Procedure

1. Create a data migration instance.
2. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
3. Configure a task name.

   DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
4. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.
	Database Type	Select **Redis**.
	Instance Mode	The value of this parameter is set to **Standalone** and cannot be changed to Cluster.
	Hostname or IP Address	Enter the endpoint that is used to connect to the source database.
	Port Number	Enter the service port number of the source database. The default port number is **6379**.
	Database Password	Enter the password of the source database. If password verification is disabled for the source database, you do not need to enter the password. **Note** After you specify the source database parameters, click **Test Connectivity** next to **Database Password** to verify whether the specified parameters are valid. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the source database parameters based on the check results.
Destination Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select **Redis**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the destination database.
	Port Number	Enter the service port number of the destination database. The default port number is **6379**.

Section	Parameter	Description
	Database Password	Enter the password of the destination database. If password verification is disabled for the destination database, you do not need to enter the password. **Note** After you specify the destination database parameters, click **Test Connectivity** next to **Database Password** to verify whether the specified parameters are valid. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the destination database parameters based on the check results.

5. In the lower-right corner of the page, click **Set Whitelist and Next**.
6. If a whitelist is configured for the user-created Redis database, you must manually add the CIDR blocks of DTS servers to the whitelist of the database. To obtain the CIDR blocks of DTS servers, click **Copy to Clipboard** in the dialog box that appears. After you add the CIDR blocks of DTS servers to the whitelist of the database, click **Next**.

   Note If you do not need to configure a whitelist for the user-created Redis database, ignore the preceding settings and click **Next**.

7. Select the migration types and the objects to be migrated.

Setting	Description
Select the migration types	○ To perform only full data migration, select only **Full Data Migration**. ○ To ensure service continuity during data migration, select both **Full Data Migration** and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.

Setting	Description
Select the objects to be migrated	Select one or more objects from the **Available** section and click the icon to move the objects to the **Selected** section. **Note** • You can select columns, tables, or databases as the objects to be migrated. • By default, after an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. • If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.
Specify the retry time for failed connection to the source or destination database	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 12 hours. You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data migration task. Otherwise, the data migration task fails. **Note** When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

8. In the lower-right corner of the page, click **Precheck**.

   Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

9. After the task passes the precheck, click **Next**.

   Note If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.3.15. Migrate data between user-created MongoDB databases

This topic describes how to migrate data between user-created MongoDB databases by using Data Transmission Service (DTS).

Prerequisites

The version of the source MongoDB database is 3.0, 3.2, 3.4, 3.6, 4.0, or 4.2.

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- If you need to migrate incremental data from a standalone MongoDB database, you must enable oplog. A standalone MongoDB database contains only a primary node. For more information, see Preparation for a standalone MongoDB database.
- If the source database uses the sharded cluster architecture, you must configure a data migration task for each shard.

Migration types

Migration type	Description
Full data migration	DTS migrates historical data of the required objects from the source database to the destination database. **Note** The following types of objects are supported: database, collection, and index.
Incremental data migration	After full data migration is complete, DTS migrates incremental data from the source database to the destination database in real time. **Note** • The create and delete operations that are performed on databases, collections, and indexes can be migrated. • The create, delete, and update operations that are performed on documents can be migrated.

Preparation for a standalone MongoDB database

If you need to migrate incremental data from a standalone MongoDB database, you must enable oplog. A standalone MongoDB database contains only a primary node. If you do not need to perform incremental data migration, skip the following steps.

Note To enable oplog, you must restart the MongoDB service. We recommend that you enable oplog during off-peak hours.

1. Use Mongo Shell to connect to the source database.
2. Run the following commands to shut down the MongoDB service:

```
use admin
db.shutdownServer()
```

3. Run the following command to start the MongoDB service from the backend as a replica set:

```
mongod --port 27017 --dbpath /var/lib/mongodb --logpath /var/log/mongodb/mongod.log --replSet rs0
--bind_ip 0.0.0.0 --auth --fork
```

> **Note**
> - The database path used by the preceding command is */var/lib/mongodb*. The log file path is */var/log/mongodb/mongod.log*. You must specify the paths based on your needs.
> - The command uses `0.0.0.0` as the associated IP address of the MongoDB service. This allows you to access the database by using all IP addresses. After the migration is complete, run the `kill` command to end the process, and start the MongoDB service by using the original configuration file.
> - The command enables authentication. You can access the database only after you pass the authentication.

4. Use Mongo Shell to connect to the source database again.
5. Run the following commands to initialize the replica set:

```
use admin
rs.initiate()
```

6. Wait until the role of the current node changes to primary, which indicates that oplog is enabled.

> **Note** You can run the `rs.printReplicationInfo()` command to view the status of oplog.

Procedure

1. Create a data migration instance.
2. Find the data migration instance that you created, and click **Configure Migration Task** in the Actions column.
3. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.
	Database Type	Select **MongoDB**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the source database.
	Port Number	Enter the service port number of the source database.
	Database Name	Enter the name of the authentication database. The database account is created in this database.
	Database Account	Enter the account of the source database. The account must have the read permissions on the source database, admin database, and local database.

Section	Parameter	Description
	Database Password	Enter the password of the source database account. **Note** After you specify the source database parameters, click **Test Connectivity** next to **Database Password** to verify whether the specified parameters are valid. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the source database parameters based on the check results.
Destination Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	The destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select **MongoDB**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the destination database.
	Port Number	Enter the service port number of the destination database.
	Database Name	Enter the name of the authentication database. The database account is created in this database. **Note** If the database account is root, enter admin.
	Database Account	Enter the account of the destination database. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account. **Note** After you specify the destination database parameters, click **Test Connectivity** next to **Database Password** to verify whether the specified parameters are valid. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the destination database parameters based on the check results.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. If a whitelist is configured for the user-created MongoDB database, you must manually add the CIDR blocks of DTS servers to the whitelist of the database. To obtain the CIDR blocks of DTS servers, click **Copy to Clipboard** in the dialog box that appears. After you add the CIDR blocks of DTS servers to the whitelist of the database, click **Next**.

   Note If you do not need to configure a whitelist for the user-created MongoDB database, ignore the preceding settings and click **Next**.

6. Select the migration types and the objects to be migrated.

Setting	Description
Select the migration types	◦ To perform only full data migration, select only **Full Data Migration**. ◦ To ensure service continuity during data migration, select both **Full Data Migration** and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.
Select the objects to be migrated	Select one or more objects from the **Available** section and click the › icon to move the objects to the **Selected** section. **Note** ◦ You can select columns, tables, or databases as the objects to be migrated. ◦ By default, after an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. ◦ If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.
Specify the retry time for failed connection to the source or destination database	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 12 hours. You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data migration task. Otherwise, the data migration task fails. **Note** When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

7. In the lower-right corner of the page, click **Precheck**.

   Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

8. After the task passes the precheck, click **Next**.

   Note If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

18.1.3.4. Manage data migration tasks

18.1.3.4.1. Object name mapping

Data Transmission Service (DTS) provides the object name mapping feature. You can use this feature to change the names of one or more objects that are migrated to the destination instance. This topic describes how to use the object name mapping feature when you configure a data migration task.

Limits

You can use the object name mapping feature only when a data migration task is configured and the current step is **Configure Migration Types and Objects**.

> **Note** Do not use the object name mapping feature after a data migration task is started. Otherwise, data may fail to be migrated.

Procedure

1. In the **Configure Migration Types and Objects** step, move the required objects to the **Selected** section, move the pointer over a database or table, and then click **Edit**.
2.
3.
4.

18.1.3.4.2. Specify an SQL condition to filter data

This topic describes how to specify an SQL condition to filter the data of a specific table when you configure a data migration task.

The SQL condition takes effect only within the table that you select. DTS migrates only the data that meets the SQL condition to the destination database. This feature is applicable to scenarios such as regular data migration and table partitioning.

Limits

An SQL condition applies only to full data migration. If you select **incremental data migration** as the migration type, the SQL condition does not filter incremental data.

Specify an SQL condition

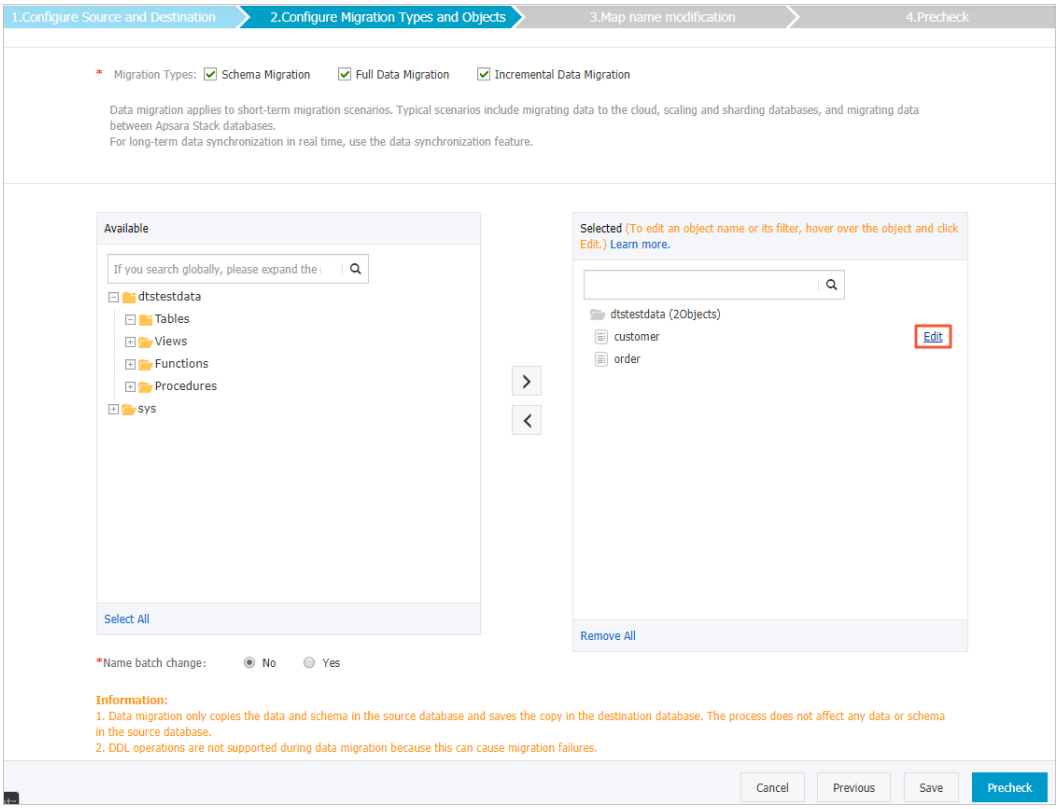
You can specify an SQL condition in the **Configure Migration Types and Objects** step when you configure a data migration task.

To filter the data of a specific table by using an SQL condition, you must select the table as the object that you want to migrate. You cannot select a database as the object. To specify an SQL condition, perform the following steps.

Procedure

1. In the **Configure Migration Types and Objects** step, move the pointer over a table in the **Selected** section. The **Edit** button appears, as shown in Edit button.

   Edit button



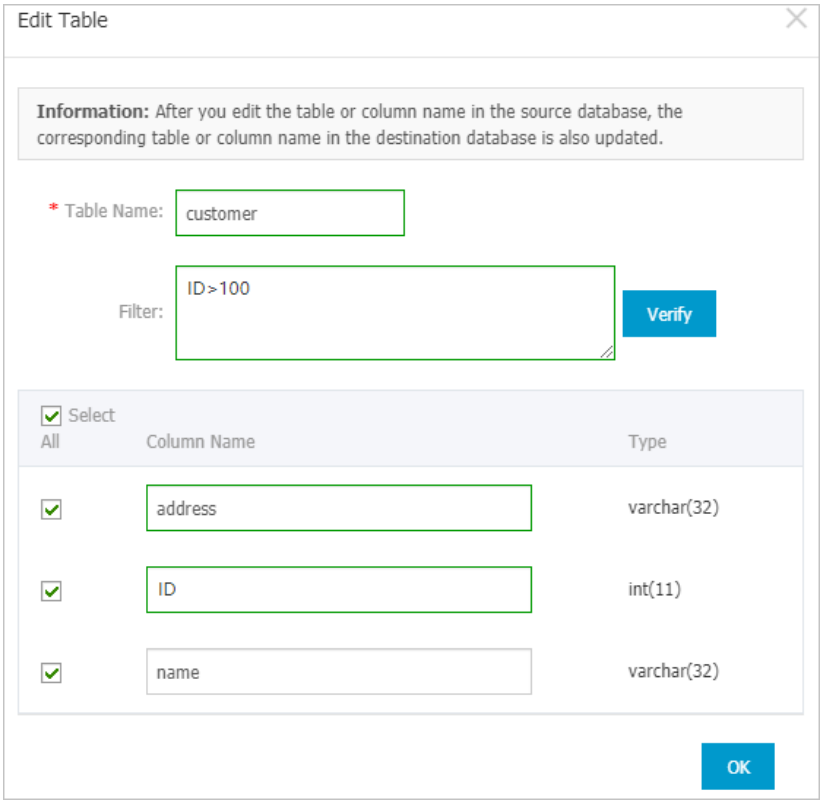
2. Click **Edit**. The Edit Table dialog box appears.

Modify an SQL condition

The SQL conditions in DTS are the same as the standard SQL WHERE conditions for databases. You can use SQL conditions to perform operations and run basic functions.

Enter an SQL condition in the text box. For example, you can enter id>1000 to migrate the records whose IDs are greater than 1,000 to the destination instance, as shown in Modify an SQL condition.

Modify an SQL condition



After the SQL condition is specified, click **OK**.

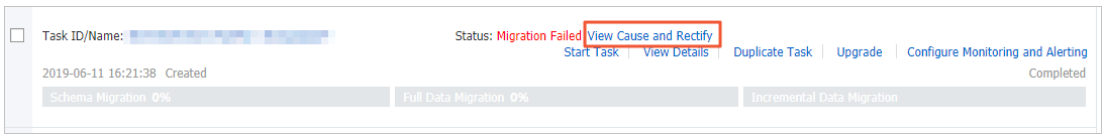
18.1.3.4.3. Troubleshoot a failed data migration task

This topic describes how to troubleshoot a failed data migration task. You can use this feature if your data migration task is in the **Migration Failed** state during schema migration or full data migration.

Troubleshoot a failed task during schema migration

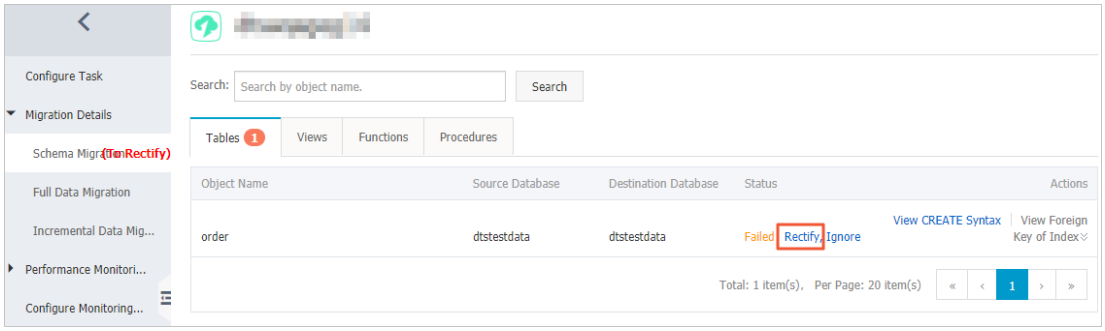
DTS supports data migration between heterogeneous data sources. However, if you migrate data of unsupported types to the destination instance during schema migration, the task may fail.

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. Use one of the following methods to troubleshoot the failed task:
   - Method 1
     a. Find the task and click **View Cause and Rectify**.



     b. Troubleshoot the issue based on the cause that is displayed in the View Cause and Rectify message. For example, you can troubleshoot an issue by modifying the schema syntax.
     c. Click **Restart Task**.
   - Method 2
     a. Click the instance ID or task name.
     b. In the left-side navigation pane, choose **Migration Details > Schema Migration**.

c. On the **Schema Migration** page, find the object that causes the migration failure and click **Rectify** in the Status column.



d. Troubleshoot the issue based on the cause that is displayed in the **Rectify** dialog box. For example, you can troubleshoot an issue by modifying the schema syntax.

e. Click **Rectify**.

> **Note**
> - If the failure persists, the **Rectify** dialog box does not close and shows the failure cause. You must continue troubleshooting based on the failure cause until the troubleshooting is successful.
> - If the troubleshooting is successful, the **Schema Migration** page appears and the status of the object changes to **Finished**.

4. If no objects are in the Failed state, DTS proceeds with the data migration task, for example, entering the full data migration process.

Troubleshoot a failed task during full data migration

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. Find the task and click **View Cause and Rectify**.

   DTS allows you to troubleshoot a task that fails during full data migration due to the following reasons.

   > **Note** If a task fails during full data migration due to other reasons, DTS provides only the **Ignore** option. The object that causes the failure is not migrated to the destination database.

   - The connection to the source or destination database failed or timed out.

     Troubleshoot the issue, make sure that the connection is successful, and then click **Restart Task**.
   - The storage space of the destination instance is insufficient or the instance is locked.

     Upgrade the specification of the destination instance or clear the log space, and then click **Restart Task**.
   - MyISAM tables in the source database are corrupted.

     Troubleshoot the issue in the source database, and then click **Restart Task**.
4. In the dialog box that appears, troubleshoot the issue based on the failure cause.
5. Click **Restart Task**.

18.1.3.5. Precheck items

18.1.3.5.1. Source database connectivity

DTS checks whether DTS servers can connect to the source database. DTS creates a connection to the source database by using the JDBC protocol. If the connection fails, the data migration task fails to pass the connectivity check.

The migration task may fail to pass the connectivity check because of the following reasons:

- The database account or password that is specified in the data migration task is invalid.

  Troubleshooting:

  Find a server that can connect to the source database. On the server, enter the database account and password that are specified in the data migration task to check whether the account and password are valid. If the database account or password is invalid, the following error message is displayed: Access deny.

  Solution:

  Log on to the DTS console, modify the database account and password, and then run a precheck again.

- The DTS servers are disallowed to access the source database.

  Troubleshooting:

  - Find a server that can connect to the source database. On the server, enter the database account and password that are specified in the data migration task to check whether the connection is successful. Only authorized DTS servers can connect to the source database. If the CIDR block of a DTS server is not included in the whitelist of the source database, the DTS server cannot connect to the source database.
  - If the source database is a MySQL database, use a MySQL client to connect to the database and run the `SELECT HOST FROM mysql.user WHERE user='Account', password='Password';` command. If the CIDR blocks of DTS servers are not included in the whitelist of the source database, the query result of the preceding command is not %.

  Solution:

  - If the source database is a MySQL database, run the `GRANT ALL ON . TO 'Account'@''%' IDENTIFIED BY 'Password';` command to authorize the database account. Replace Account and Password in the preceding command with your database account and password. After the account is authorized, run a precheck again.

- A firewall is configured on the source database server.

  Troubleshooting: If the server where the source database resides runs Linux, run the `iptables -L` command in the shell to check whether a firewall is configured for the server. If the server where the source database resides runs Windows, find Windows Defender Firewall from the Control Panel and check whether a firewall is configured for the server.

  Solution:

  Disable the firewall and run a precheck again.

- The network between DTS servers and the source database is unavailable.

  If the failure persists, you can check whether the network between DTS servers and the source database is available. In this case, we recommend that you contact Alibaba Cloud engineers by submitting a ticket.

18.1.3.5.2. Check the destination database connectivity

This check item checks whether the DTS server can connect to the destination database for migration. DTS creates a connection to the destination database by using the JDBC protocol. If the connection fails, the check item fails.

The destination database connectivity precheck may fail for the following reasons:

- An incorrect account or password is provided when a migration task is created.

  Diagnostics:

On any network-ready server that can connect to the destination database, use the account and password specified for creating the migration task to connect to the destination database through client software. Check whether the connection succeeds. If an error is reported for the connection and the error message contains Access deny, the account or password is incorrect.

Troubleshooting:

Modify the migration task in the DTS console, correct the account and password, and perform the precheck again.

- There is no connectivity between the DTS server and destination database.

  If you check that the password and account are correct, the check item may fail because there is no connectivity between the DTS server and the destination database. In this case, contact the DTS engineers on duty.

18.1.3.5.3. Binary logging configurations of the source database

Whether binary logging is enabled in the source database

This item is checked only when you migrate incremental data between MySQL databases. DTS checks whether binary logging is enabled in the source database. If binary logging is disabled in the source database, the check result is Failed.

Troubleshooting: Run the `log_bin=mysql_bin` command to modify the configuration file of the source database. Restart the source database and run a precheck again.

Binary log format of the source database

This item is checked only when you migrate incremental data between MySQL databases. DTS checks whether the binary log format is set to ROW in the source database. If the binary log format is not set to ROW in the source database, the check result is Failed.

Troubleshooting: Run the `set global binlog_format=ROW` command in the source database and run a precheck again. We recommend that you restart the MySQL process. Otherwise, data loss may occur because sessions will continue to be written in a non-ROW mode.

Binary log files in the source database

This item is checked only when you migrate incremental data between MySQL databases. DTS checks whether specific binary log files are removed from the source database. If binary log files in the source database are incomplete, the check result is Failed.

Troubleshooting: Run the `PURGE BINARY LOGS TO 'The name of the first binary log file that is not deleted'` command in the source database and run a precheck again.

To find the binary log files that are removed from the source database, click the info icon next to the failed item. In the **View Details** dialog box, the names of deleted binary log files are displayed.

Parameter binlog_row_image of the source database

This item is checked only when you migrate incremental data between MySQL databases. DTS checks whether the value of the binlog_row_image parameter in the source database is set to FULL. This parameter indicates whether the full image is recorded. If the full image is not recorded in binary log files of the source database, the check result is Failed.

Troubleshooting: Run the `set global binlog_row_image=FULL` command in the source database and run a precheck again.

18.1.3.5.4. Integrity of the FOREIGN KEY constraints

DTS checks whether the parent table on which a child table depends is included in the selected objects. The precheck allows DTS to protect the integrity of the FOREIGN KEY constraints.

If the parent table on which a child table depends is not included in the selected objects, the check result is Failed.

Troubleshooting:

- Do not migrate the child tables that cause the check failure. To do this, remove these child tables from the selected objects and run a precheck again.
- Migrate the parent tables rather than the child tables. To do this, add the parent tables to the selected objects and run a precheck again.
- Delete the foreign key dependencies between the parent and child tables in the source database and run a precheck again.

18.1.3.5.5. Existence of FEDERATED tables

This item is checked only when you migrate incremental data between MySQL databases. DTS checks whether the source database contains storage engines that are not supported by incremental data migration. Incremental data migration does not support the FEDERATED and MRG_MyISAM storage engines.

If **the FEDERATED storage engine is used by specific tables in the source database**, the check result is Failed.

If **the MRG_MyISAM storage engine is used by specific tables in the source database**, the check result is Failed.

Solution:

Remove the tables that use the FEDERATED or MRG_MyISAM storage engine from the selected objects. Then, create a separate migration task to perform schema migration and full data migration for these tables.

18.1.3.5.6. Permissions

Source database permissions

DTS checks whether the account of the source database has the required permissions to perform data migration. For information about the permissions that are required by each type of database, see the topics about how to configure data migration tasks.

Destination database permissions

DTS checks whether the account of the destination database has the required permissions to perform data migration. For information about the permissions that are required by each type of database, see the topics about how to configure data migration tasks.

18.1.3.5.7. Object name conflict

This check item checks for duplicate object names in the destination and source database. If this check item fails, an object in the destination RDS instance has the same name as an object to be migrated. This causes the migration to fail.

When this check item fails, an error message is displayed indicating that an object in the destination database has the same name as an object to be migrated from the source database.

Troubleshooting:

- Use the database and table name mapping feature provided by DTS to migrate the object to be migrated to another object with a different name in the destination database.

- In the destination database, delete or rename the object that has the same name as the object to be migrated.
- Modify the migration task and delete that object to be migrated from the list of objects to be migrated. Do not migrate this object.

18.1.3.5.8. Schema existence

This check item checks whether the database to be migrated exists in the destination RDS instance. If no, DTS creates one automatically. However, under the following circumstances, the automatic database creation fails, and this check item prompts a failure:

- The database name contains characters other than lowercase letters, digits, underscores (_), and hyphens (-).

  The cause of the precheck failure is that the name of the **source database** does not comply with the requirements of RDS.

  Troubleshooting: On the database management page of the RDS console, create a database that complies with the requirements of RDS and grant the migration account the read and write permissions on the new database. Use the database name mapping feature provided by DTS to map the source database to the new database. Then, perform the precheck again.

- The character set of the database is not UTF8, GBK, Latin1, or UTF-8MB4.

  The cause of the precheck failure is that the character set of the **source database** does not comply with the requirements of RDS.

  Troubleshooting: On the database management page of the RDS console, create a database that complies with the requirements of RDS and grant the migration account the read and write permissions on the new database. If the new database and the database to be migrated have different names, you can use the database name mapping feature of DTS to map the database to be migrated to the new database. Then re-run the precheck.

- The migration account of the destination database has no read and write permissions on the database to be migrated.

  The cause of the precheck failure is that you are not authorized to operate on the **source database**.

  Troubleshooting: On the database management page of the RDS console, click the Account Management tab. Grant the migration account the read and write permissions on the source database. Then, perform the precheck again.

18.1.3.5.9. Value of server_id in the source database

This item is checked only when you migrate incremental data between MySQL databases. DTS checks whether the value of **server-id** in the source database is set to an integer greater than 1.

If the check result is Failed, run the `set global server_id='An integer greater than 1'` command in the source database and perform a precheck again.

18.1.3.5.10. Source database version

DTS checks whether the version of the source database is supported. The table Source database types and versions lists the source database versions that are supported by DTS.

Source database types and versions

Source database type	Supported version
MySQL	5.0, 5.1, 5.5, 5.6, and 5.7. Only 5.1, 5.5, 5.6, and 5.7 are supported for incremental data migration.

If the check result is Failed, you must upgrade or downgrade the source database to a supported version before you perform a precheck again.

18.1.3.6. Data type mappings between heterogeneous databases

Heterogeneous databases support different data types. During schema migration, Data Transmission Service (DTS) converts the data types of the source database into those of the destination database. This topic lists the data type mappings for you to evaluate the impact of data migration on your business.

Data migration from a user-created Oracle database to a user-created MySQL database or an ApsaraDB RDS for MySQL instance

Data type in the Oracle database	Data type in the MySQL database	Supported by DTS
varchar2(n [char/byte])	varchar(n)	Yes
nvarchar2[(n)]	national varchar[(n)]	Yes
char[(n [byte/char])]	char[(n)]	Yes
nchar[(n)]	national char[(n)]	Yes
number[(p[,s])]	decimal[(p[,s])]	Yes
float(p)]	double	Yes
long	longtext	Yes
date	datetime	Yes
binary_float	decimal(65,8)	Yes
binary_double	double	Yes
timestamp[(fractional_seconds_precision)]	datetime[(fractional_seconds_precision)]	Yes
timestamp[(fractional_seconds_precision)]with localtimezone	datetime[(fractional_seconds_precision)]	Yes
timestamp[(fractional_seconds_precision)]with localtimezone	datetime[(fractional_seconds_precision)]	Yes
clob	longtext	Yes
nclob	longtext	Yes
blob	longblob	Yes
raw	varbinary(2000)	Yes
long raw	longblob	Yes
bfile	N/A	No
interval year(year_precision) to month	N/A	No
interval day(day_precision)to second[(fractional_seconds_precision)]	N/A	No

Note

- A char column with a length greater than 255 bytes is converted to the varchar(n) type.
- Data types such as bfile, interval year to month, and interval day to second in Oracle databases are not supported in MySQL databases. They cannot be converted to data types supported by the destination database during schema migration.

  The schema migration fails if the table to be migrated contains these three data types. You must make sure that columns with these three data types are excluded from the objects to be migrated.
- The timestamp data type of MySQL databases does not contain the time zone information. However, the timestamp with time zone and timestamp with local time zone data types in Oracle databases provide time zone information. Therefore, DTS converts the values of these data types based on the time zone to UTC time for storage in the destination instance.

Data migration from a user-created Oracle database to a PolarDB-X instance

Oracle data type	PolarDB-X data type	Supported by DTS
varchar2(n [char/byte])	varchar(n)	Yes
nvarchar2[(n)]	national varchar[(n)]	Yes
char[(n [byte/char])]	char[(n)]	Yes
nchar[(n)]	national char[(n)]	Yes
number[(p[,s])]	decimal[(p[,s])]	Yes
float(p)]	double	Yes
long	longtext	Yes
date	datetime	Yes
binary_float	decimal(65,8)	Yes
binary_double	double	Yes
timestamp[(fractional_seconds_precision)]	datetime[(fractional_seconds_precision)]	Yes
timestamp[(fractional_seconds_precision)]with localtimezone	datetime[(fractional_seconds_precision)]	Yes
timestamp[(fractional_seconds_precision)]with localtimezone	datetime[(fractional_seconds_precision)]	Yes
clob	longtext	Yes
nclob	longtext	Yes
blob	longblob	Yes
raw	varbinary(2000)	Yes
long raw	longblob	Yes

Oracle data type	PolarDB-X data type	Supported by DTS
bfile	None	No
interval year(year_precision) to month	None	No
interval day(day_precision)to second[(fractional_seconds_precision)]	None	No

Note

- If a char field in the Oracle database is greater than 255 bytes in length, DTS converts this field to the varchar(n) type in the PolarDB-X instance.
- The timestamp data type of PolarDB-X does not contain the time zone information. However, the timestamp with time zone and timestamp with local time zone data types in Oracle databases provide the time zone information. Therefore, DTS converts the values of these data types into UTC time in the destination PolarDB-X instance.

Data migration from a user-created Oracle database to a PolarDB cluster

Oracle data type	PolarDB data type	Supported by DTS
varchar2(n [char/byte])	varchar2[(n)]	Yes
nvarchar2[(n)]	nvarchar2[(n)]	Yes
char[(n [byte/char])]	char[(n)]	Yes
nchar[(n)]	nchar[(n)]	Yes
number[(p[,s])]	number[(p[,s])]	Yes
float(p)]	double precision	Yes
long	long	Yes
date	date	Yes
binary_float	real	Yes
binary_double	double precision	Yes
timestamp[(fractional_seconds_precision)]	timestamp[(fractional_seconds_precision)]	Yes
timestamp[(fractional_seconds_precision)]with time zone	timestamp[(fractional_seconds_precision)]with time zone	Yes
timestamp[(fractional_seconds_precision)]with local time zone	timestamp[(fractional_seconds_precision)]with time zone	Yes
clob	clob	Yes
nclob	nclob	Yes
blob	blob	Yes

Oracle data type	PolarDB data type	Supported by DTS
raw	raw(size)	Yes
long raw	long raw	Yes
bfile	None	No
interval year(year_precision) to month	interval year to month	No
interval day(day_precision) to second[(fractional_seconds_precision)]	interval day to second[(fractional_seconds_precision)]	No

Note PolarDB does not support the timestamp[(fractional_seconds_precision)]with local time zone data type. DTS converts the data of this type into UTC time and then stores the data in the destination PolarDB cluster by using the timestamp[(fractional_seconds_precision)]with time zone data type.

18.1.4. Data synchronization

18.1.4.1. Database types, initial synchronization types, and synchronization topologies

You can use Data Transmission Service (DTS) to synchronize data between various data sources. This topic describes the database types, initial synchronization types, and synchronization topologies that are supported by DTS.

Source database	Destination database	Initial synchronization type	Synchronization topology
• User-created MySQL database 5.1, 5.5, 5.6, and 5.7 • RDS MySQL	User-created MySQL database 5.1, 5.5, 5.6, and 5.7	Initial schema synchronization Initial full data synchronization	One-way synchronization Two-way synchronization
	RDS MySQL 5.6 and 5.7	Initial schema synchronization Initial full data synchronization	One-way synchronization Two-way synchronization
	AnalyticDB for MySQL 2.0 and 3.0	Initial schema synchronization Initial full data synchronization	One-way synchronization

5.6 and 5.7 Source database	Destination database	Initial synchronization type	Synchronization topology
	AnalyticDB for PostgreSQL 4.3 and 6.0	Initial schema synchronization Initial full data synchronization	One-way synchronization
	Datahub	Initial schema synchronization	One-way synchronization
	MaxCompute	Initial schema synchronization Initial full data synchronization	One-way synchronization
Cloud Native Distributed Database PolarDB-X (formerly known as DRDS)	Cloud Native Distributed Database PolarDB-X	Initial full data synchronization	One-way synchronization
	Datahub	Initial schema synchronization	One-way synchronization
	AnalyticDB for MySQL 2.0 and 3.0	Initial schema synchronization Initial full data synchronization	One-way synchronization

18.1.4.2. Create a data synchronization instance

Before you configure a task to synchronize data, you must create a data synchronization instance. This topic describes how to create a data synchronization instance in the Data Transmission Service (DTS) console.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Synchronization**.
3. In the upper-right corner of the page, click **Create Synchronization Task**.
4. In the Create DTS Instances dialog box, set the required parameters.

Parameter	Description
Source Instance Region	Select the region where the source instance resides.
Source Instance Type	Select the type of the source instance. ○ **MySQL**: a user-created MySQL database or an ApsaraDB RDS for MySQL instance ○ **Drds**: a Cloud Native Distributed Database PolarDB-X instance (formerly known as DRDS)

Parameter	Description
Destination Instance Region	Select the region where the destination instance resides.
Destination Instance Type	Select the type of the destination instance. ○ **MySQL**: a user-created MySQL database or an ApsaraDB RDS for MySQL instance ○ **AnalyticDB**: AnalyticDB for MySQL ○ **MaxCompute** ○ **DataHub** ○ **Drds**: a Cloud Native Distributed Database PolarDB-X instance (formerly known as DRDS) ○ **AnalyticDB for PostgreSQL**: AnalyticDB for PostgreSQL
Synchronization Mode	Two-way synchronization is available only when you select **MySQL** as the type of both the source and destination instances.
Instances to Create	Set the number of data synchronization instances that you want to create at a time. The default value is 1.

Note In the Create DTS Instances dialog box, you can view the total number of instances, the number of existing instances, and the number of instances that can be created.

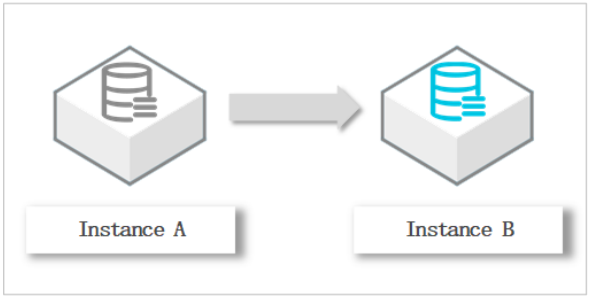
5. Click **Create**.

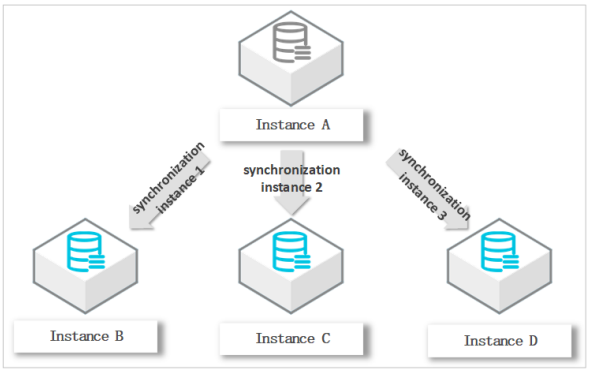
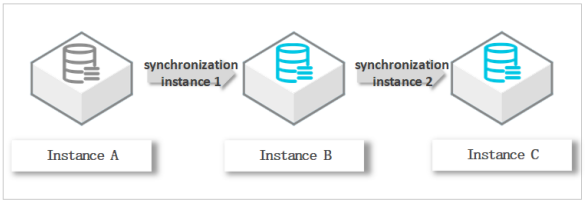
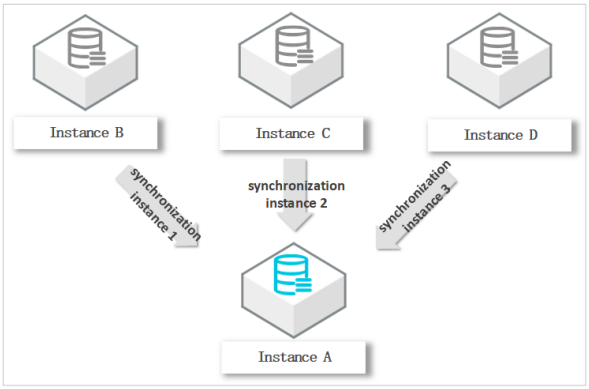
18.1.4.3. Synchronization topologies

The data synchronization feature supports multiple types of synchronization topologies. You can select a topology for your data synchronization instances based on your business requirements. This topic describes the synchronization topologies that are supported by DTS and how to use these topologies.

One-way synchronization

To ensure data consistency for one-way synchronization, we recommend that you perform only read operations on the objects in the destination instance. Do not modify the objects.

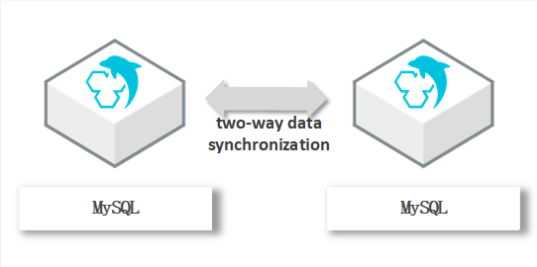
Topology type	Topology	Description
One-way one-to-one synchronization		None

Topology type	Topology	Description
One-way one-to-many synchronization		You must purchase multiple synchronization instances to implement one-way one-to-many synchronization. For example, if you want to synchronize data from Instance A to Instance B, C, and D, you must purchase three synchronization instances.
One-way cascade synchronization		You must purchase multiple synchronization instances to implement one-way cascade synchronization. For example, if you want to synchronize data from Instance A to Instance B and then from Instance B to Instance C, you must purchase two synchronization instances.
One-way many-to-one synchronization		You must purchase multiple synchronization instances to implement one-way many-to-one synchronization. For example, if you want to synchronize data from Instance B, C, and D to Instance A, you must purchase three synchronization instances. **Note** To ensure data consistency, you must select different objects for these synchronization instances.

Two-way synchronization

DTS supports two-way data synchronization only between two MySQL databases. DTS does not support two-way data synchronization between multiple MySQL databases.

Topology type	Topology	Description

Topology type	Topology	Description
Two-way one-to-one synchronization		To ensure data consistency, make sure that records with the same primary key, business primary key, or unique key are updated only on one of the instances.

18.1.4.4. Configure data synchronization tasks

18.1.4.4.1. Configure data synchronization between ApsaraDB RDS for MySQL instances

This topic describes how to configure one-way data synchronization between ApsaraDB RDS for MySQL instances.

Prerequisites

The source and destination ApsaraDB RDS for MySQL instances are created.

Precautions

- DTS uses the read and write resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- If you select one or more tables (not a database) as the required objects, do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.
-
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After initial full data synchronization, the tablespace of the destination instance is larger than that of the source instance.

Supported synchronization topologies

DTS supports one-way synchronization and two-way synchronization. For more information, see Synchronization topologies.

SQL operations that can be synchronized

Limits

- Incompatibility with triggers

  If you select a database as the object and the database contains a trigger that updates a table, data inconsistency may occur. To solve this issue, you must delete the trigger in the destination database.

- Limits on RENAME TABLE operations

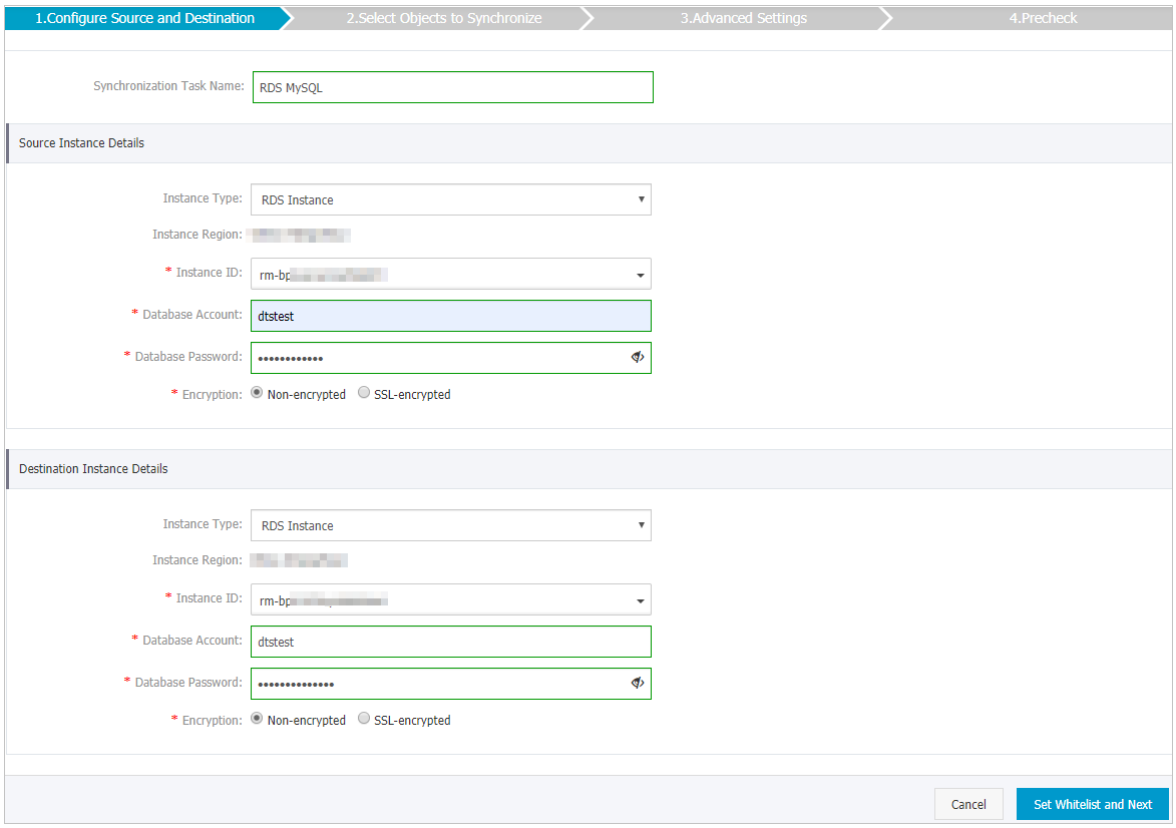
  RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if you select a table as the object and rename the table during data synchronization, the data of this table is not synchronized to the destination database. To avoid this situation, you can select the database to which this table belongs as the object when you configure the data synchronization task.

Procedure

1. Create a data synchronization instance.

   > **Note** When you create the data synchronization instance, set both Source Instance Type and Destination Instance Type to **MySQL**, and set Synchronization Mode to **One-Way Synchronization**.

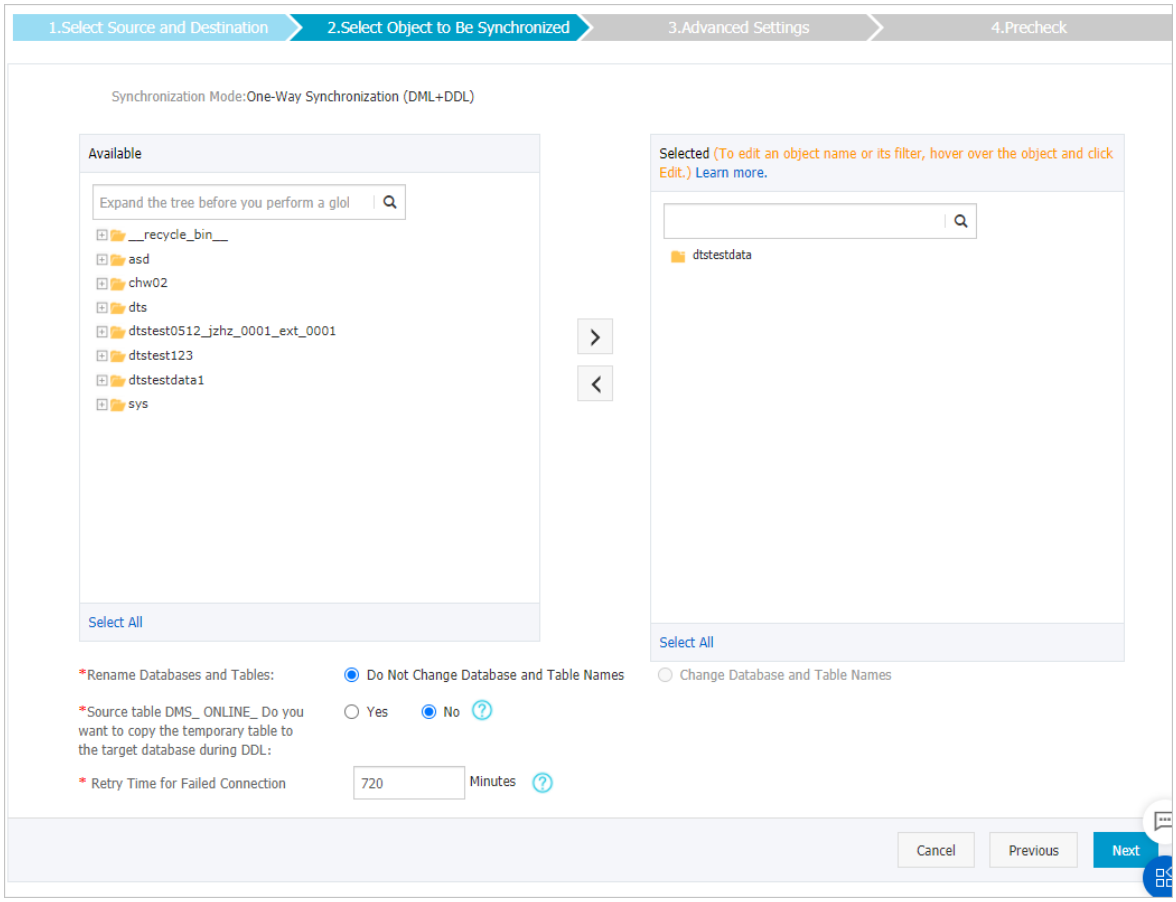
2. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.
3. Configure the source and destination instances.



Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
	Instance Type	Select **RDS Instance**.

Section	Parameter	Description
Source Instance Details	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Instance ID	Select the ID of the source RDS instance.
	Database Account	Enter the database account of the source RDS instance. **Note** If the database engine of the source RDS instance is **MySQL 5.6**, you do not need to configure the **database account** or **database password**.
	Database Password	Enter the password of the source database account.
	Encryption	Select an encryption method. If you select **SSL-encrypted**, you must enable SSL encryption for the RDS instance before you configure the data synchronization task.
Destination Instance Details	Instance Type	Select **RDS Instance**.
	Instance Region	The region of the source instance. The region is the same as the destination region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Instance ID	Select the ID of the destination RDS instance.
	Database Account	Enter the database account of the destination RDS instance. **Note** If the database engine of the destination RDS instance is **MySQL 5.6**, you do not need to configure the **database account** or **database password**.
	Database Password	Enter the password of the destination database account.
	Encryption	Select an encryption method. If you select **SSL-encrypted**, you must enable SSL encryption for the RDS instance before you configure the data synchronization task.

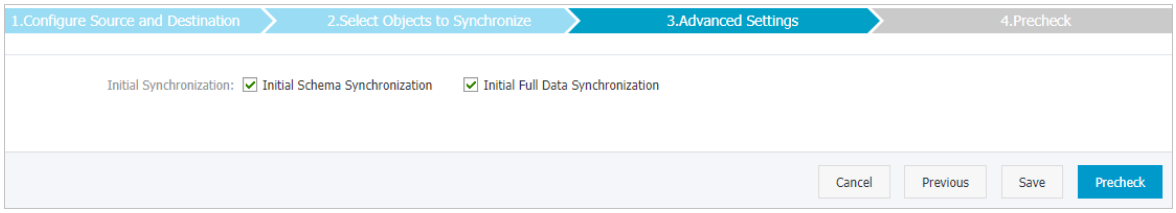
4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Select the processing mode of conflicting tables, and the objects that you want to synchronize.



Parameter	Description
Processing Mode In Existed Target Table	◦ **Pre-check and Intercept**: checks whether the destination database contains tables that have the same names as tables in the source database. If the source and destination databases do not contain identical table names, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. ◦ **Ignore**: skips the precheck for identical table names in the source and destination databases. **Warning** If you select **Ignore**, data consistency is not guaranteed and your business may be exposed to potential risks. ▪ DTS does not synchronize the data records that have the same primary keys as the data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. ▪ If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only specific columns are synchronized or the data synchronization task fails.

Parameter	Description
Select Objects	Select objects (tables or a database) from the **Available** section and click the icon to move the objects to the **Selected** section. **Note** • If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. • After an object is synchronized to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are synchronized to the destination instance. For more information, see Specify the name of an object in the destination instance.

6. In the lower-right corner of the page, click **Next**.
7. Configure initial synchronization.



Note Initial synchronization includes initial schema synchronization and initial full data synchronization. If you select both **Initial Schema Synchronization** and **Initial Full Data Synchronization**, DTS synchronizes the schemas and historical data of the required objects before DTS synchronizes incremental data.

8. In the lower-right corner of the page, click **Precheck**.

Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

9. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.** Then, DTS performs initial synchronization.

18.1.4.4.2. Synchronize data from an ApsaraDB RDS for MySQL instance to a MaxCompute project

MaxCompute (formerly known as ODPS) is a fast and fully managed computing platform for large-scale data warehousing. MaxCompute can process exabytes of data. This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to a MaxCompute project by using Data Transmission Service (DTS).

Precautions

- DTS uses the read and write resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend

that you synchronize data during off-peak hours.

- If you select one or more tables (not a database) as the required objects, do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.
- You can select only tables as the objects to be synchronized.
- MaxCompute does not support the PRIMARY KEY constraint. If network errors occur, DTS may synchronize duplicate data records to MaxCompute.

SQL operations that can be synchronized

- Data definition language (DDL) operation: ADD COLUMN
- Data manipulation language (DML) operations: INSERT, UPDATE, and DELETE

Synchronization process

1. Initial schema synchronization

   DTS synchronizes the schemas of the required objects from the source database to MaxCompute. During initial schema synchronization, DTS adds the _base suffix to the end of the source table name. For example, if the name of the source table is customer, the name of the table in MaxCompute is customer_base.

2. Initial full data synchronization

   DTS synchronizes the historical data of the table from the source database to the destination table in MaxCompute. For example, the customer table in the source database is synchronized to the customer_base table in MaxCompute. The data is the basis for subsequent incremental synchronization.

   > **Note** The destination table that is suffixed with _base is known as a full baseline table.

3. Incremental data synchronization

   DTS creates an incremental data table in MaxCompute. The name of the incremental data table is suffixed with _log, for example, customer_log. Then, DTS synchronizes the incremental data that was generated in the source database to the incremental data table.

   > **Note** For more information, see Schema of an incremental data table.

Procedure

1. Create a data synchronization instance.

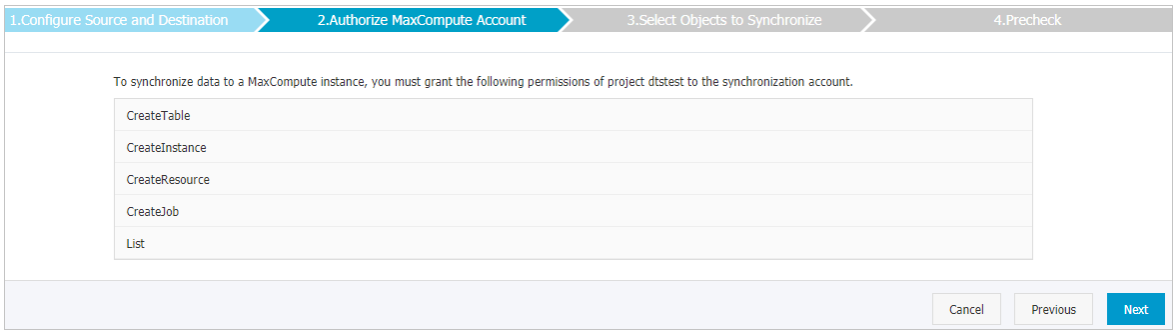
   > **Note** When you create the data synchronization instance, set Source Instance Type to **MySQL**, set Destination Instance Type to **MaxCompute**, and set Synchronization Mode to **One-Way Synchronization**.

2. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.
3. Configure the source and destination instances.

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.

Section	Parameter	Description
Source Instance Details	Instance Type	Select **RDS Instance**.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Instance ID	Select the ID of the source RDS instance.
	Database Account	Enter the database account of the source RDS instance. **Note** If the database engine of the source RDS instance is **MySQL 5.5** or **MySQL 5.6**, you do not need to configure the **database account** or **database password**.
	Database Password	Enter the password of the source database account.
	Encryption	Select **Non-encrypted** or **SSL-encrypted**. If you select **SSL-encrypted**, you must enable SSL encryption for the RDS instance before you configure the data synchronization task.
Destination Instance Details	Instance Type	This parameter is set to **MaxCompute** and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Project	The name of the MaxCompute project.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. In the lower-right corner of the page, click **Next**. In this step, the permissions on the MaxCompute project are granted to the synchronization account.



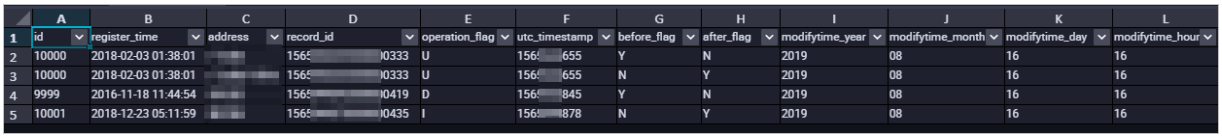
6. Configure the synchronization policy and objects.
7. In the lower-right corner of the page, click **Precheck**.

Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

8. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.** Then, DTS performs initial synchronization.

Schema of an incremental data table

DTS synchronizes incremental data that is generated in the source MySQL database to the incremental data table in MaxCompute. The incremental data table stores incremental data and specific metadata. The following figure shows the schema of an incremental data table.



	A	B	C	D	E	F	G	H	I	J	K	L
1	id	register_time	address	record_id	operation_flag	utc_timestamp	before_flag	after_flag	modifytime_year	modifytime_month	modifytime_day	modifytime_hour
2	10000	2018-02-03 01:38:01	[redacted]	1565[redacted]0333	U	156[redacted]655	Y	N	2019	08	16	16
3	10000	2018-02-03 01:38:01	[redacted]	1565[redacted]0333	U	156[redacted]655	N	Y	2019	08	16	16
4	9999	2016-11-18 11:44:54	[redacted]	1565[redacted]0419	D	156[redacted]845	Y	N	2019	08	16	16
5	10001	2018-12-23 05:11:59	[redacted]	1565[redacted]0435	I	156[redacted]878	N	Y	2019	08	16	16

> **Note** In the example, the `modifytime_year`, `modifytime_month`, `modifytime_day`, `modifytime_hour`, and `modifytime_minute` fields form the partition key.

The following table describes the schema of an incremental data table.

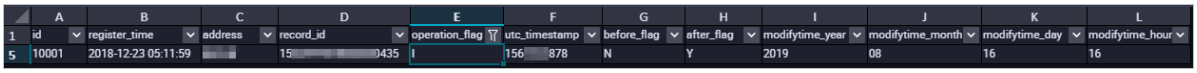
Field	Description
record_id	The ID of the incremental log entry. **Note** • The ID auto-increments for each new log entry. • If an UPDATE operation is performed, DTS generates two incremental log entries for the operation. The two incremental log entries have the same record ID.
operation_flag	The operation type. Valid values: • I: an INSERT operation. • D: a DELETE operation. • U: an UPDATE operation.
utc_timestamp	The operation timestamp. It is also the timestamp of the binary log file. The timestamp is in the UTC format.
before_flag	Indicates whether the column values are pre-update values. Valid values: Y and N.
after_flag	Indicates whether the column values are post-update values. Valid values: Y and N.

Additional information about the before_flag and after_flag fields

For different operation types, the **before_flag** and **after_flag** fields of an incremental log entry are defined as follows:

- INSERT

  For an INSERT operation, the column values are the newly inserted record values (post-update values). The value of the before_flag field is N and the value of the after_flag field is Y.

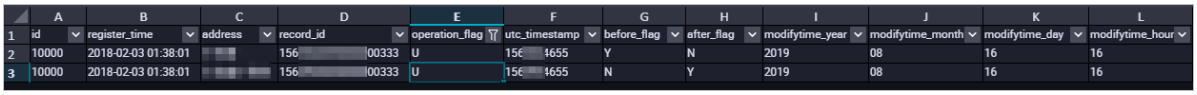


	A	B	C	D	E	F	G	H	I	J	K	L
1	id	register_time	address	record_id	operation_flag	utc_timestamp	before_flag	after_flag	modifytime_year	modifytime_month	modifytime_day	modifytime_hour
5	10001	2018-12-23 05:11:59	[redacted]	15[redacted]0435	I	156[redacted]878	N	Y	2019	08	16	16

- UPDATE

  DTS generates two incremental log entries for an UPDATE operation. The two incremental log entries have the same values for the record_id, operation_flag, and dts_utc_timestamp fields.

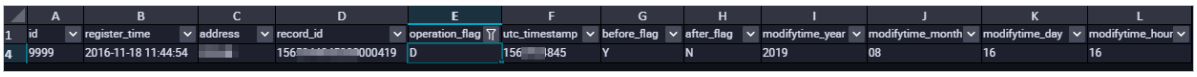
  The second log entry records the pre-update values, so the value of the before_flag field is Y and the value of the after_flag field is N. The second log entry records the post-update values, so the value of the before_flag field is N and the value of the after_flag field is Y.



	A	B	C	D	E	F	G	H	I	J	K	L
1	id	register_time	address	record_id	operation_flag	utc_timestamp	before_flag	after_flag	modifytime_year	modifytime_month	modifytime_day	modifytime_hour
2	10000	2018-02-03 01:38:01	[redacted]	156[redacted]00333	U	15[redacted]4655	Y	N	2019	08	16	16
3	10000	2018-02-03 01:38:01	[redacted]	156[redacted]00333	U	15[redacted]4655	N	Y	2019	08	16	16

- DELETE

  For a DELETE operation, the column values are the deleted record values (pre-update values). The value of the before_flag field is Y and the value of the after_flag field is N.

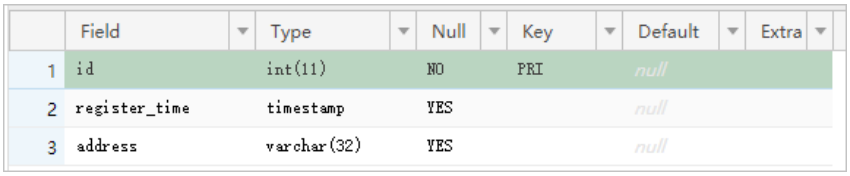


	A	B	C	D	E	F	G	H	I	J	K	L
1	id	register_time	address	record_id	operation_flag	utc_timestamp	before_flag	after_flag	modifytime_year	modifytime_month	modifytime_day	modifytime_hour
4	9999	2016-11-18 11:44:54	[redacted]	156[redacted]000419	D	156[redacted]845	Y	N	2019	08	16	16

Merge a full baseline table and incremental data table

After a data synchronization task is started, DTS creates a full baseline table and an incremental data table in MaxCompute. You can use SQL statements to merge the two tables. This allows you to obtain the full data at a specific time point.

This section describes how to merge data for the customer table. The following figure shows the schema of the customer table.



	Field	Type	Null	Key	Default	Extra
1	id	int(11)	NO	PRI	null	
2	register_time	timestamp	YES		null	
3	address	varchar(32)	YES		null	

1. Create a table in MaxCompute based on the schema of the source table. The table is used to store the merged data.

   For example, you can obtain full data of the customer table at the `1565944878` time point. Run the following SQL statements to create the required table:

   ```
   CREATE TABLE `customer_1565944878` (
       `id` bigint NULL,
       `register_time` datetime NULL,
       `address` string);
   ```

   > **Note** For more information about the data types that are supported by MaxCompute, see MaxCompute User Guide.

2. Run the following SQL statements in MaxCompute to merge the full baseline table and incremental data table and obtain full data at a specific time point:

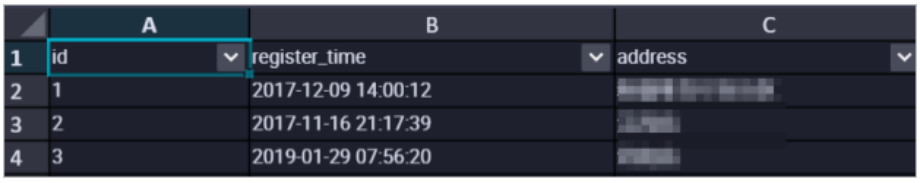
```
set odps.sql.allow.fullscan=true;
insert overwrite table <result_storage_table>
select <col1>,
       <col2>,
       <colN>
  from(
select row_number() over(partition by t.<primary_key_column>
 order by record_id desc, after_flag desc) as row_number, record_id, operation_flag, after_flag,
<col1>, <col2>, <colN>
  from(
select incr.record_id, incr.operation_flag, incr.after_flag, incr.<col1>, incr.<col2>,incr.<colN>
  from <table_log> incr
 where utc_timestamp< <timestamp>
 union all
select 0 as record_id, 'I' as operation_flag, 'Y' as after_flag, base.<col1>, base.<col2>,base.<c
olN>
  from <table_base> base) t) gt
where record_num=1
  and after_flag='Y'
```

> **Note**
>
> - <result_storage_table>: the name of the table that stores the merged data.
> - <col1>/<col2>/<colN>: the names of the columns in the table to be merged.
> - <primary_key_column>: the name of the primary key column in the table to be merged.
> - <table_log>: the name of the incremental data table.
> - <table_base>: the name of the full baseline table.
> - <timestamp>: the timestamp that is generated when full data is obtained.

Run the following SQL statements to obtain full data of the customer table at the `1565944878` time point:

```
set odps.sql.allow.fullscan=true;
insert overwrite table customer_1565944878
select id,
       register_time,
       address
  from(
select row_number() over(partition by t.id
 order by record_id desc, after_flag desc) as row_number, record_id, operation_flag, after_flag,
id, register_time, address
  from(
select incr.record_id, incr.operation_flag, incr.after_flag, incr.id, incr.register_time, incr.ad
dress
  from customer_log incr
 where utc_timestamp< 1565944878
 union all
select 0 as record_id, 'I' as operation_flag, 'Y' as after_flag, base.id, base.register_time, bas
e.address
  from customer_base base) t) gt
 where gt.row_number= 1
   and gt.after_flag= 'Y';
```

3. Query the merged data from the customer_1565944878 table.



	A	B	C
1	id	register_time	address
2	1	2017-12-09 14:00:12	
3	2	2017-11-16 21:17:39	
4	3	2019-01-29 07:56:20	

18.1.4.4.3. Synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for MySQL cluster

AnalyticDB for MySQL is a real-time online analytical processing (RT-OLAP) service that is developed by Alibaba Cloud for online data analysis with high concurrency. This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for MySQL cluster by using Data Transmission Service (DTS). After you synchronize data, you can use AnalyticDB for MySQL to build internal business intelligence (BI) systems, interactive query systems, and real-time report systems.

Prerequisites

- The tables that you want to synchronize from the ApsaraDB RDS for MySQL instance contain primary keys.
- The destination AnalyticDB for MySQL cluster has sufficient storage space.

Precautions

- DTS uses the read and write resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- If you select one or more tables (not a database) as the required objects, do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.
- If the disk space usage of nodes in an AnalyticDB for MySQL cluster reaches 80%, the cluster is locked. We recommend that you estimate the required disk space based on the objects to be synchronized. You must ensure that the destination cluster has sufficient storage space.

SQL operations that can be synchronized

- Data definition language (DDL) operations: CREATE TABLE, DROP TABLE, RENAME TABLE, TRUNCATE TABLE, ADD COLUMN, and DROP COLUMN
- Data manipulation language (DML) operations: INSERT, UPDATE, and DELETE

> **Note** We recommend that you do not change the data type of fields in the source table during data synchronization. Otherwise, DTS generates an error message and stops the data synchronization task.

Data type mappings

The data types of ApsaraDB RDS for MySQL and AnalyticDB for MySQL do not have one-to-one correspondence. During initial schema synchronization, DTS converts the data types of the source database into those of the destination database. The following table lists the data types that DTS can convert.

Data type of ApsaraDB RDS for MySQL	Data type of AnalyticDB for MySQL
BIGINT UNSIGNED	DECIMAL(20,0)
BINARY	VARBINARY
BIT	VARCHAR

Data type of ApsaraDB RDS for MySQL	Data type of AnalyticDB for MySQL
BLOB	VARBINARY
CHAR	VARCHAR
DATE	DATE
DATETIME	DATETIME
DECIMAL	DECIMAL
DOUBLE	DOUBLE
ENUM	VARCHAR
FLOAT	FLOAT
GEOMETRY	VARBINARY
GEOMETRYCOLLECTION	VARBINARY
INT UNSIGNED	BIGINT
INTEGER	INT
JSON	JSON
LINESTRING	VARBINARY
LONGBLOB	VARBINARY
LONGTEXT	VARCHAR
MEDIUMBLOB	VARBINARY
MEDIUMINT	INT
MEDIUMINT UNSIGNED	INT
MEDIUMTEXT	VARCHAR
MULTILINESTRING	VARBINARY
MULTIPOINT	VARBINARY
MULTIPOLYGON	VARBINARY
NUMBERIC	DECIMAL
POINT	VARBINARY
POLYGON	VARBINARY
SET	VARCHAR
SMALLINT UNSIGNED	INT
TEXT	VARCHAR

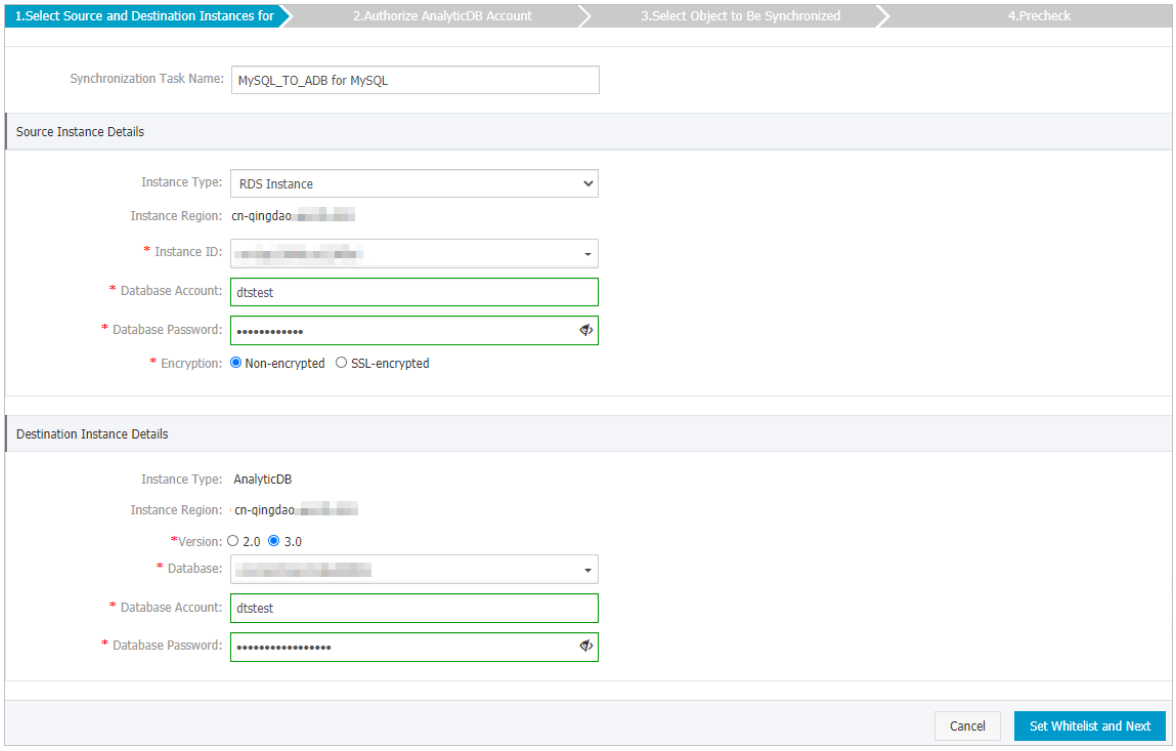
Data type of ApsaraDB RDS for MySQL	Data type of AnalyticDB for MySQL
TIME	TIME
TIMESTAMP	TIMESTAMP
TINYBLOB	VARBINARY
TINYINT UNSIGNED	SMALLINT
TINYTEXT	VARCHAR
VARBINARY	VARBINARY
VARCHAR	VARCHAR
YEAR	INT

Procedure

1. Create a data synchronization instance.

   > **Note** When you create the data synchronization instance, set Source Instance Type to **MySQL**, set Destination Instance Type to **AnalyticDB**, and set Synchronization Mode to **One-Way Synchronization**.

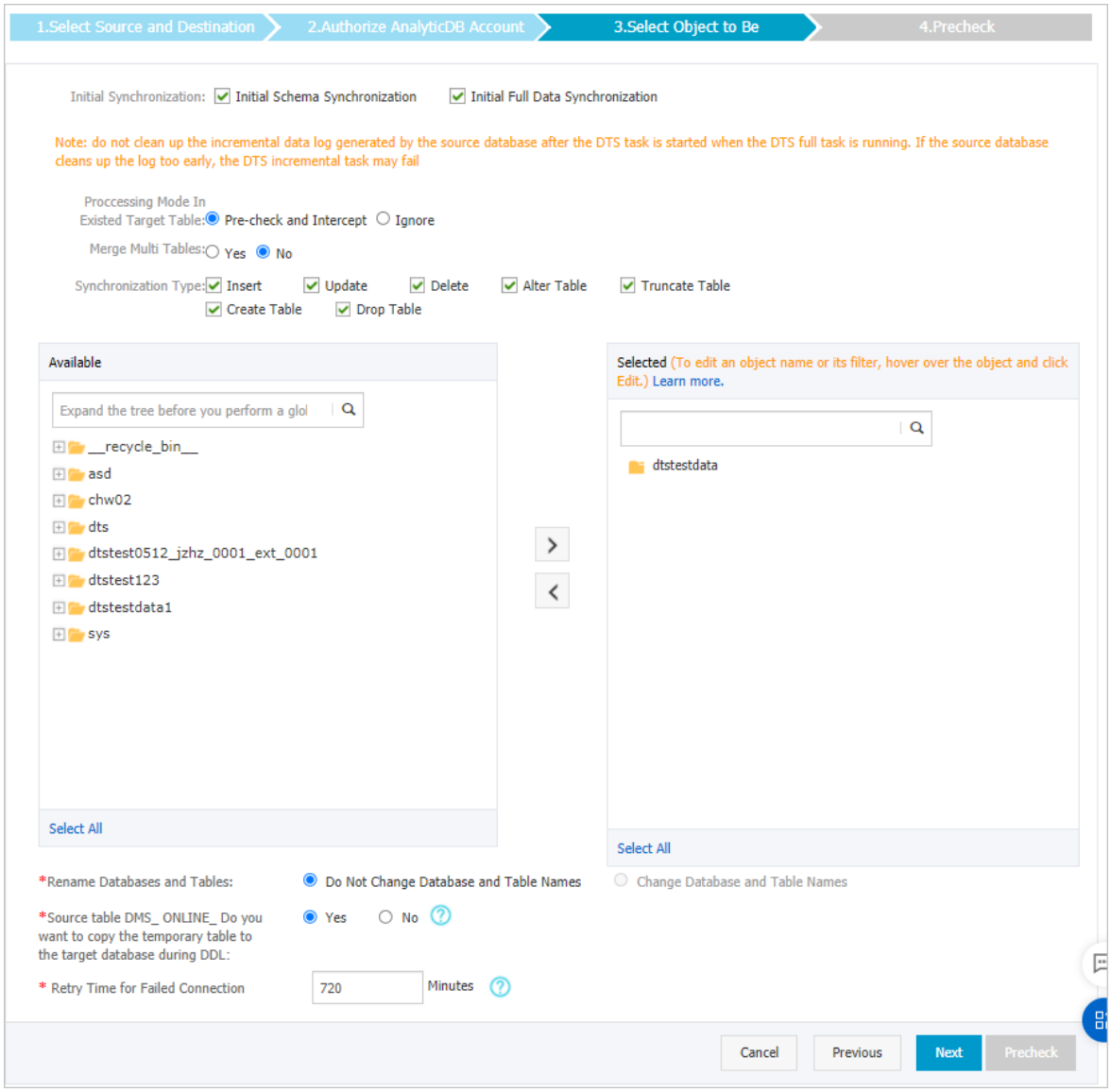
2. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.
3. Configure the source and destination instances.



Section	Parameter	Description

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **RDS Instance**.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Instance ID	Select the ID of the source RDS instance.
	Database Account	Enter the database account of the source RDS instance. The account must have the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. **Note** If the database engine of the source RDS instance is **MySQL 5.5** or **MySQL 5.6**, you do not need to configure the **database account** or **database password**.
	Database Password	Enter the password of the source database account.
	Encryption	Select an encryption method. If you select **SSL-encrypted**, you must enable SSL encryption for the RDS instance before you configure the data synchronization task.
Destination Instance Details	Instance Type	This parameter is set to **AnalyticDB** and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Version	Select **3.0**.
	Database	Select the ID of the destination AnalyticDB for MySQL cluster.
	Database Account	Enter the database account of the destination AnalyticDB for MySQL cluster. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Configure the synchronization policy and objects.

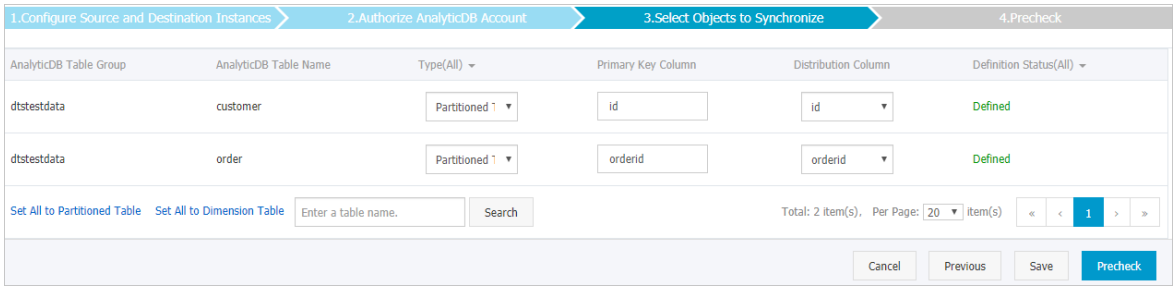


Parameter	Description
Initial Synchronization	You must select both **Initial Schema Synchronization** and **Initial Full Data Synchronization** in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination cluster. The schemas and data are the basis for subsequent incremental synchronization.

Parameter	Description
Processing Mode In Existed Target Table	◦ **Pre-check and Intercept**: checks whether the destination database contains tables that have the same names as tables in the source database. If the source and destination databases do not contain identical table names, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. ◦ **Ignore**: skips the precheck for identical table names in the source and destination databases. **Warning** If you select **Ignore**, data consistency is not guaranteed and your business may be exposed to potential risks. ▪ DTS does not synchronize the data records that have the same primary keys as the data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. ▪ If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only specific columns are synchronized or the data synchronization task fails.
Merge Multi Tables	◦ If you select **Yes**, DTS adds the `__dts_data_source` column to each table to record data sources. In this case, DDL operations cannot be synchronized. ◦ **No** is selected by default. In this case, DDL operations can be synchronized. **Note** You can merge the data source columns based on tasks rather than tables. To merge only the data source columns of specific tables, you can create two data synchronization tasks.
Synchronization Type	Select the types of operations that you want to synchronize based on your business requirements. All operation types are selected by default. **Note** Only the INSERT, UPDATE, DELETE, and ADD COLUMN operations can be synchronized.

Parameter	Description
Select Objects	Select objects (tables or a database) from the **Available** section and click the icon to move the objects to the **Selected** section. **Note** • If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. • After an object is synchronized to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are synchronized to the destination instance. For more information, see Specify the name of an object in the destination instance.

6. In the lower-right corner of the page, click **Next**.
7. Specify a type for the tables that you want to synchronize to the destination database.



Note After you select **Initial Schema Synchronization**, you must specify the **type**, **primary key column**, and **partition key column** for the tables that you want to synchronize to AnalyticDB for MySQL.

8. In the lower-right corner of the page, click **Precheck**.

Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

9. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.** Then, DTS performs initial synchronization.

18.1.4.4.4. Synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for PostgreSQL instance

This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for PostgreSQL instance by using Data Transmission Service (DTS). The data synchronization feature allows you to transfer and analyze data with ease.

Prerequisites

- The tables that you want to synchronize contain primary keys.
- An AnalyticDB for PostgreSQL instance is created.

Precautions

- If you select one or more tables (not a database) as the required objects, do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- Only one-way synchronization is supported.

Limits

- You can select only tables as the objects to be synchronized.
- DTS does not synchronize the schemas of the required objects from the source database to the destination database.
- DTS does not synchronize the following types of data: JSON, GEOMETRY, CURVE, SURFACE, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, GEOMETRYCOLLECTION, and BYTEA.

SQL operations that can be synchronized

- Data manipulation language (DML) operations: INSERT, UPDATE, and DELETE
- Data definition language (DDL) operations: ALTER TABLE, ADD COLUMN, DROP COLUMN, and RENAME COLUMN

> **Note** The CREATE TABLE and DROP TABLE operations are not supported. To synchronize data from a new table, you must add the table to the selected objects. For more information, see Add objects to be synchronized.

Term mappings

Term in ApsaraDB RDS for MySQL	Term in AnalyticDB for PostgreSQL
Database	Schema
Table	Table

Create a data structure in the destination instance

Create a database, schema, and table in the destination AnalyticDB for PostgreSQL instance based on the data structure of the source RDS instance.

Configure a data synchronization task

1. Create a data synchronization instance.

> **Note** When you create the data synchronization instance, set Source Instance Type to **MySQL**, set Destination Instance Type to **AnalyticDB for PostgreSQL**, and set Synchronization Mode to **One-Way Synchronization**.

2. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.
3. Configure the source and destination instances.

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **RDS Instance**.
	Instance Region	The region where the source RDS instance resides.
	Instance ID	Select the ID of the source RDS instance.
	Encryption	Select **Non-encrypted** or **SSL-encrypted**. **Note** If you select **SSL-encrypted**, you must enable SSL encryption for the RDS instance before you configure the data synchronization task.
Destination Instance Details	Instance Type	This parameter is set to **AnalyticDB for PostgreSQL** and cannot be changed.
	Instance Region	The region where the destination instance resides.
	Instance ID	Select the ID of the destination AnalyticDB for PostgreSQL instance.
	Database Name	Enter the name of the destination database.
	Database Account	Enter the database account of the destination AnalyticDB for PostgreSQL instance. **Note** The database account must have the SELECT, INSERT, UPDATE, DELETE, COPY, TRUNCATE, and ALTER TABLE permissions.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Wait until the synchronization account is created. Then, click **Next**.
6. Configure the synchronization policy and objects.

Section	Parameter	Description
Synchronization policy	Initial Synchronization	Select **Initial Full Data Synchronization**. **Note** DTS synchronizes the historical data of the required objects from the source instance to the destination instance. The historical data is the basis for subsequent incremental synchronization.
	Processing Mode In Existed Target Table	◦ **Pre-Check and intercept** (Selected by default) Checks the **Schema Name Conflict** item and generates an error message if the destination table contains data. ◦ **Clear Target Table** Skips the **Schema Name Conflict** item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. ◦ **Ignore** Skips the **Schema Name Conflict** item during the precheck. Adds data to the existing data during initial full data synchronization. If you want to synchronize data from multiple tables to one table, you can select this mode.
	Synchronization Type	◦ **Insert** ◦ **Update** ◦ **Delete** ◦ **AlterTable** **Note** Select the types of operations that you want to synchronize based on your business requirements.
Select Objects	N/A	You can select only tables as the objects to be synchronized. You can use the object name mapping feature to change the names of the columns that are synchronized to the destination database. For more information, see Specify the name of an object in the destination instance. **Note** The CREATE TABLE operation is not supported. To synchronize data from a new table, you must add the table to the selected objects. For more information, see Add objects to be synchronized.

7. In the lower-right corner of the page, click **Precheck**.

   > **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

8. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.** Then, DTS performs initial synchronization.

18.1.4.4.5. Synchronize data between Cloud Native Distributed Database PolarDB-X instances

Cloud Native Distributed Database PolarDB-X is formerly known as Distributed Relational Database Service (DRDS). It is compatible with the MySQL protocol and syntax, and supports automatic sharding, online smooth scaling, auto scaling, and transparent read/write splitting. This topic describes how to synchronize data between Cloud Native Distributed Database PolarDB-X instances by using Data Transmission Service (DTS).

Prerequisites

The tables that you want to synchronize contain primary keys.

Precautions

- DTS uses the read and write resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- We recommend that you do not change the network type of the Cloud Native Distributed Database PolarDB-X instances during data synchronization.

  > **Note** After you change the network type of a Cloud Native Distributed Database PolarDB-X instance during data synchronization, you must submit a ticket to resume the data synchronization instance.

- We recommend that you do not scale up or down the databases in the Cloud Native Distributed Database PolarDB-X instances. Otherwise, data may fail to be synchronized.

Supported synchronization topologies

DTS supports the following synchronization topologies: one-way one-to-one synchronization, one-way one-to-many synchronization, one-way cascade synchronization, and one-way many-to-one synchronization. For more information, see Synchronization topologies.

SQL operations that can be synchronized

The INSERT, UPDATE, and DELETE operations can be synchronized.

Before you begin

Create a database and tables in the destination instance based on the schemas of the objects in the source instance. This is because DTS does not support **initial schema synchronization** between Cloud Native Distributed Database PolarDB-X instances.

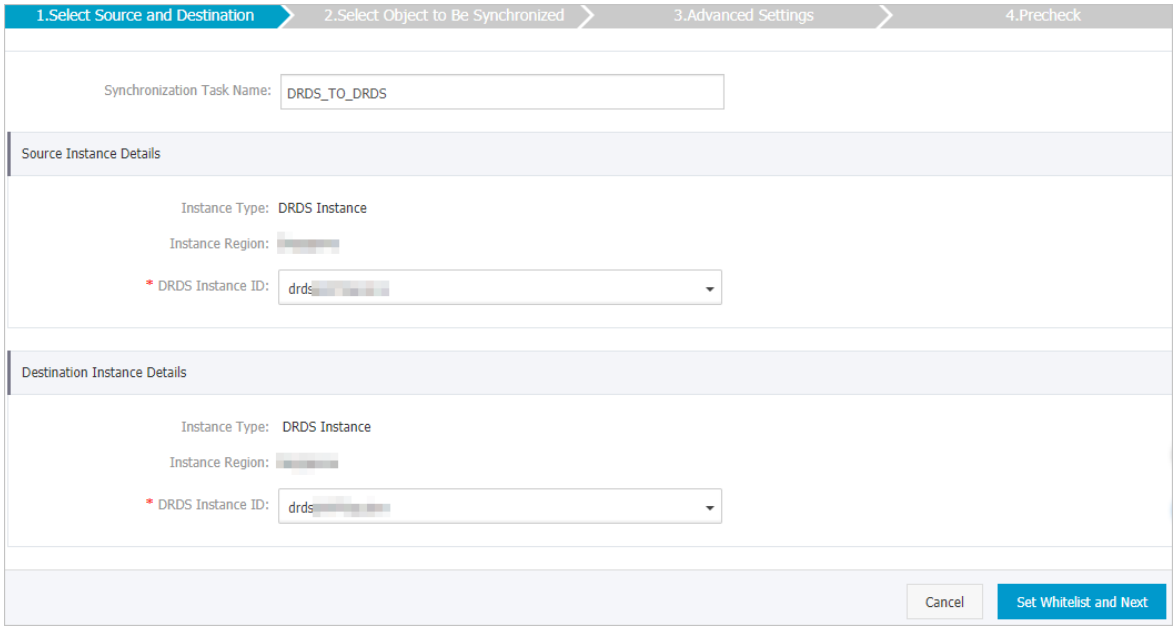
> **Note** During **initial schema synchronization**, DTS synchronizes the schemas of the required objects from the source database to the destination database.

Procedure

1. Create a data synchronization instance.

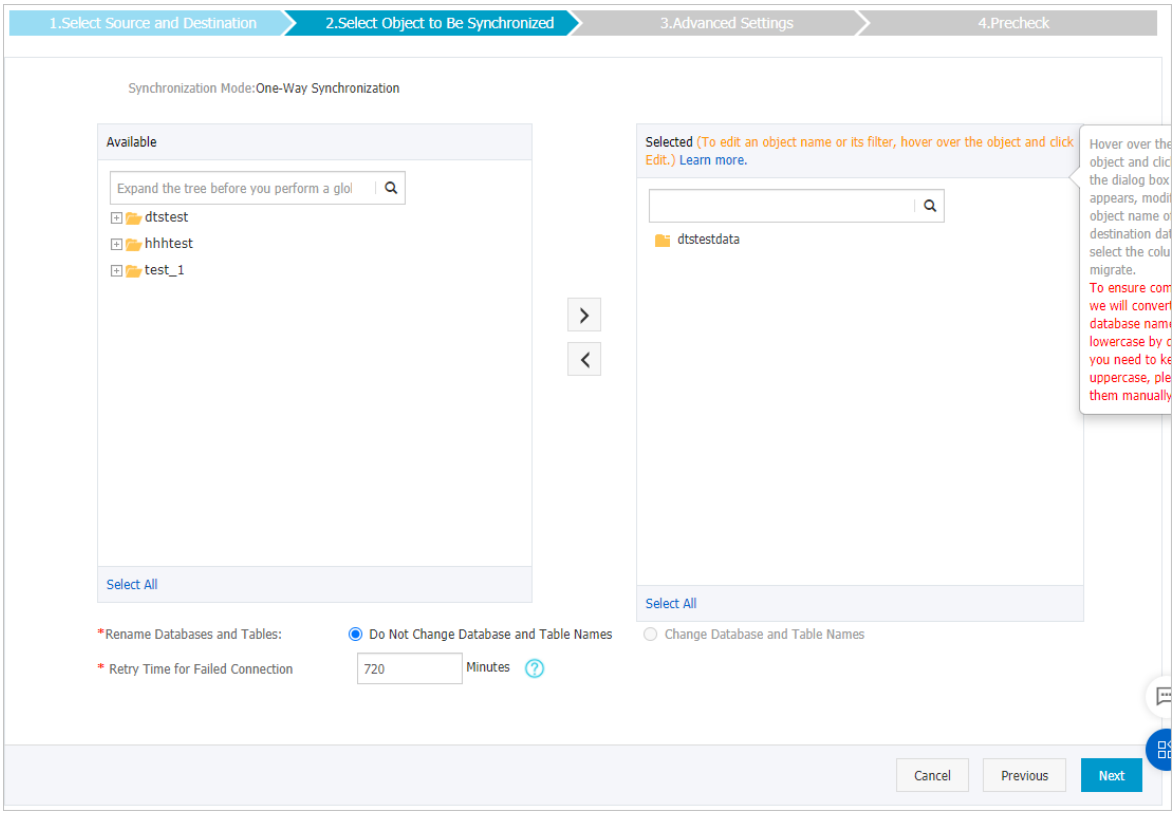
   > **Note** When you create the data synchronization instance, set both Source Instance Type and Destination Instance Type to **Drds**, and set Synchronization Mode to **One-Way Synchronization**.

2. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.
3. Configure the source and destination instances.



Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	This parameter is set to **DRDS Instance** and cannot be changed.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the source Cloud Native Distributed Database PolarDB-X instance.
Destination Instance Details	Instance Type	This parameter is set to **DRDS Instance** and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the destination Cloud Native Distributed Database PolarDB-X instance.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Configure the synchronization policy and objects.



Parameter	Description
Processing Mode In Existed Target Table	○ **Pre-check and Intercept**: checks whether the destination tables are empty. If the destination tables are empty, the precheck is passed. If the tables are not empty, an error is returned during the precheck and the data synchronization task cannot be started. ○ **Ignore**: skips the check for empty destination tables. **Warning** If you select **Ignore**, data consistency is not guaranteed and your business may be exposed to potential risks. ■ If the source and destination databases have the same schema, DTS does not synchronize the data records that have the same primary keys as the data records in the destination database. ■ If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only specific columns are synchronized or the data synchronization task fails.

Parameter	Description
Select Objects	Select tables from the **Available** section and click the icon to move the tables to the **Selected** section. **Note** - You can select only tables as the objects to be synchronized. - After an object is synchronized to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are synchronized to the destination instance. For more information, see Specify the name of an object in the destination instance.

6. Click **Next**.
7. Specify whether you want to perform initial full data synchronization.

   Note During **initial full data synchronization**, DTS synchronizes the historical data of the required objects from the source database to the destination database. If you do not select Initial Full Data Synchronization, DTS does not synchronize the historical data.

8. In the lower-right corner of the page, click **Precheck**.

   Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

9. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.** Then, DTS performs initial synchronization.

18.1.4.4.6. Synchronize data from a Cloud Native Distributed Database PolarDB-X instance to an AnalyticDB for MySQL cluster

This topic describes how to synchronize data from a Cloud Native Distributed Database PolarDB-X instance to an AnalyticDB for MySQL cluster by using Data Transmission Service (DTS).

Prerequisites

The tables that you want to synchronize contain primary keys.

Precautions

- If you select one or more tables (not a database) as the required objects, do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- Only one-way synchronization is supported.

Supported synchronization topologies

DTS supports the following synchronization topologies: one-way one-to-one synchronization, one-way one-to-many synchronization, and one-way many-to-one synchronization. For more information, see Synchronization topologies.

SQL operations that can be synchronized

INSERT, UPDATE, and DELETE

Data type mappings

The data types of ApsaraDB RDS for MySQL and AnalyticDB for MySQL do not have one-to-one correspondence. During initial schema synchronization, DTS converts the data types of the source database into those of the destination database. The following table lists the data types that DTS can convert.

Data type of ApsaraDB RDS for MySQL	Data type of AnalyticDB for MySQL
BIGINT UNSIGNED	DECIMAL(20,0)
BINARY	VARBINARY
BIT	VARCHAR
BLOB	VARBINARY
CHAR	VARCHAR
DATE	DATE
DATETIME	DATETIME
DECIMAL	DECIMAL
DOUBLE	DOUBLE
ENUM	VARCHAR
FLOAT	FLOAT
GEOMETRY	VARBINARY
GEOMETRYCOLLECTION	VARBINARY
INT UNSIGNED	BIGINT
INTEGER	INT
JSON	JSON
LINESTRING	VARBINARY
LONGBLOB	VARBINARY
LONGTEXT	VARCHAR
MEDIUMBLOB	VARBINARY
MEDIUMINT	INT

Data type of ApsaraDB RDS for MySQL	Data type of AnalyticDB for MySQL
MEDIUMINT UNSIGNED	INT
MEDIUMTEXT	VARCHAR
MULTILINESTRING	VARBINARY
MULTIPOINT	VARBINARY
MULTIPOLYGON	VARBINARY
NUMBERIC	DECIMAL
POINT	VARBINARY
POLYGON	VARBINARY
SET	VARCHAR
SMALLINT UNSIGNED	INT
TEXT	VARCHAR
TIME	TIME
TIMESTAMP	TIMESTAMP
TINYBLOB	VARBINARY
TINYINT UNSIGNED	SMALLINT
TINYTEXT	VARCHAR
VARBINARY	VARBINARY
VARCHAR	VARCHAR
YEAR	INT

Procedure

1. Create a data synchronization instance.

   > **Note** When you create the data synchronization instance, set Source Instance Type to **Drds**, set Destination Instance Type to **AnalyticDB**, and set Synchronization Mode to **One-Way Synchronization**.

2. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.
3. Configure the source and destination instances.

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.

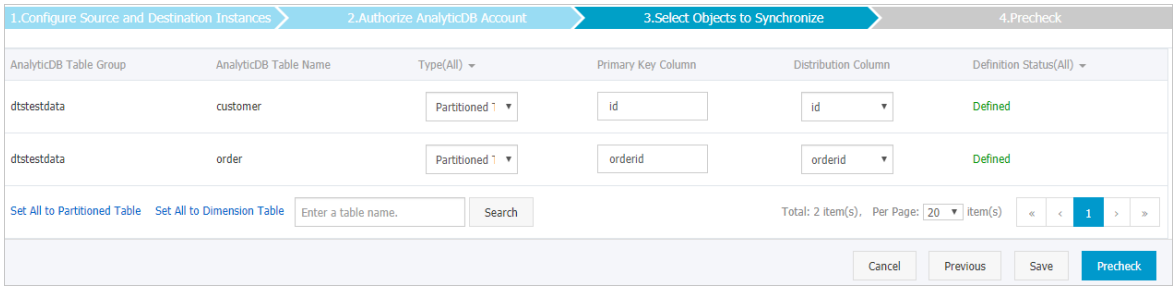
Section	Parameter	Description
Source Instance Details	Instance Type	This parameter is set to **DRDS Instance** and cannot be changed.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the source Cloud Native Distributed Database PolarDB-X instance.
Destination Instance Details	Instance Type	This parameter is set to **AnalyticDB** and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Version	Select **3.0**.
	Database	Select the ID of the destination AnalyticDB for MySQL cluster.
	Database Account	Enter the database account of the destination AnalyticDB for MySQL cluster.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Configure the synchronization policy and objects

Section	Parameter	Description
Synchronization policy	Initial Synchronization	You must select both **Initial Schema Synchronization** and **Initial Full Data Synchronization** in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination cluster. The schemas and data are the basis for subsequent incremental synchronization.
	Processing Mode In Existed Target Table	◦ **Clear Target Table** Skips the **Schema Name Conflict** item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. ◦ **Ignore** Skips the **Schema Name Conflict** item during the precheck. Adds data to the existing data during initial full data synchronization. If you want to synchronize data from multiple tables to one table, you can select this mode.

Section	Parameter	Description
	Synchronization Type	Select the types of operations that you want to synchronize based on your business requirements. ○ **Insert** ○ **Update** ○ **Delete**
Select Objects	N/A	Select tables from the **Available** section and click the icon to move the tables to the **Selected** section. **Note** ○ You can select only tables as the objects to be synchronized. ○ You can use the object name mapping feature to change the names of the columns that are synchronized to the destination database. For more information, see Specify the name of an object in the destination instance.

6. Click **Next**.
7. Specify a type for the tables that you want to synchronize to the destination database.



Note After you select **Initial Schema Synchronization**, you must specify the **type**, **primary key column**, and **partition key column** for the tables that you want to synchronize to AnalyticDB for MySQL.

8. In the lower-right corner of the page, click **Precheck**.

Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

9. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.** Then, DTS performs initial synchronization.

18.1.4.4.7. Synchronize data from a Cloud Native Distributed Database PolarDB-X instance to a DataHub instance

This topic describes how to synchronize data from a Cloud Native Distributed Database PolarDB-X instance to a DataHub instance by using Data Transmission Service (DTS). After you synchronize data, you can use big data services such as Realtime Compute to analyze data in real time.

Prerequisites

- The tables that you want to synchronize have PRIMARY KEY or UNIQUE constraints.
- A DataHub project is created to receive the synchronized data.

Precautions

- If you select one or more tables (not a database) as the required objects, do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- Only one-way synchronization is supported.

Limits

- You can select only tables as the objects to be synchronized.
- Initial full data synchronization is not supported. DTS does not synchronize the historical data of the required objects from the source PolarDB-X instance to the destination DataHub instance.
- DTS does not synchronize data definition language (DDL) operations to the destination database. If you perform a DDL operation on the source PolarDB-X instance during data synchronization, data fails to be synchronized. To solve this issue, you must modify the related topic in the destination DataHub instance and then restart the data synchronization task.

SQL operations that can be synchronized

INSERT, UPDATE, and DELETE

Procedure

1. Create a data synchronization instance.

   > **Note** When you create the data synchronization instance, set Source Instance Type to **Drds**, set Destination Instance Type to **Datahub**, and set Synchronization Mode to **One-Way Synchronization**.

2. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.
3. Configure the source and destination instances.

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	This parameter is set to **DRDS Instance** and cannot be changed.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the source Cloud Native Distributed Database PolarDB-X instance.

Section	Parameter	Description
Destination Instance Details	Instance Type	This parameter is set to **DataHub** and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Project	Select the name of the DataHub project.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Configure the synchronization policy and objects.

Parameter	Description
Initial Synchronization	Select **Initial Schema Synchronization**. **Note** After you select **Initial Schema Synchronization**, DTS synchronizes the schemas of the required objects (such as tables) to the destination DataHub instance.
Select Objects	Select objects from the **Available** section and click the › icon to move the objects to the **Selected** section. **Note** ◦ You can select only tables as the objects to be synchronized. ◦ You can use the object name mapping feature to change the names of the columns that are synchronized to the destination database. For more information, see Specify the name of an object in the destination instance.

6. In the lower-right corner of the page, click **Precheck**.

Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.** Then, DTS performs initial synchronization.

18.1.4.4.8. Configure two-way data synchronization between RDS instances

18.1.4.4.8.1. Overview

DTS supports two-way real-time data synchronization between RDS instances on any two clouds. This section describes how to use DTS to create a two-way synchronization task between two ApsaraDB RDS for MySQL instances for active geo-redundancy, geo-disaster recovery, and other scenarios.

18.1.4.4.8.2. Supported synchronization statements

Two-way synchronization between ApsaraDB RDS for MySQL instances supports all DML updates (including INSERT, UPDATE, and DELETE) and the following DDL updates:

- ALTER TABLE, ALTER VIEW, ALTER FUNCTION, and ALTER PROCEDURE
- CREATE DATABASE, CREATE SCHEMA, CREATE INDEX, CREATE TABLE, CREATE PROCEDURE, CREATE FUNCTION, CREATE TRIGGER, CREATE VIEW, and CREATE EVENT
- DROP FUNCTION, DROP EVENT, DROP INDEX, DROP PROCEDURE, DROP TABLE, DROP TRIGGER, and DROP VIEW
- RENAME TABLE and TRUNCATE TABLE

> **Note** To ensure the stability of a two-way synchronization channel, you can synchronize DDL updates on the same table in only one direction.
>
> For example, for two-way synchronization, you must enable DDL synchronization in either the **A-to-B** or **B-to-A** direction. If DDL synchronization is configured in one direction, it is not supported in the reverse direction. You can only perform DML synchronization.

18.1.4.4.8.3. Detect and resolve conflicts

To ensure data consistency, for two-way synchronized instances, make sure that records with the same primary key, business primary key, or unique key are updated only on one of the instances. If you unexpectedly update a record with the same primary key, business primary key, or unique key on both instances that are two-way synchronized, a synchronization conflict occurs. To maximize the stability of two-way synchronized instances, DTS supports detecting and resolving data conflicts.

Considerations

During two-way synchronization, the system time of the source and destination instances may not be the same. Additionally, synchronization delays may occur. For these reasons, DTS cannot guarantee that its conflict detection mechanism can completely prevent data conflicts. You must refactor certain business logic to ensure that records of the same primary key, business primary key, or unique key are updated only on one of the instances that are two-way synchronized.

Supported conflict types

Currently, DTS supports detecting the following conflict types:

- Uniqueness conflicts caused by INSERT operations

  A uniqueness conflict occurs when the synchronization of an inserted row violates the unique constraint. For example, if two instances in two-way synchronization insert a record with the same primary key value at almost the same time, one of the inserted records fails to be synchronized because a record with the same primary key value already exists in the destination instance.

- Inconsistent records caused by UPDATE operations

  Update conflicts occur in the following scenarios:

  - The records to be updated do not exist in the destination instance. If the records to be updated do not exist, DTS automatically changes the UPDATE operation to the INSERT operation and inserts these records to the destination instance. In this case, duplicate unique key values may occur.
  - The primary keys or unique keys of the records to be updated conflict with each other.

- A DELETE operation is made on non-existent records

  A delete conflict occurs when the records to be deleted do not exist in the destination instance.

  In this case, DTS automatically ignores the DELETE operation regardless of the conflict resolution policy that you have configured.

Supported conflict resolution policies

For the preceding synchronization conflicts, DTS provides the following resolution policies. You can select a conflict resolution policy as required when configuring two-way synchronization.

- TaskFailed: The synchronization task reports an error and automatically exits the process in case of a conflict.

  When the synchronization encounters a conflict of the preceding types, the synchronization task reports an error and automatically exits the process. The task enters a failed state and you must manually resolve the conflict. This method is the default conflict resolution policy.

- Ignore: The records in the destination instance are used in case of a conflict.

  When the synchronization encounters a conflict of the preceding types, the synchronization task skips the current synchronization statement and continues the process. The records in the destination instance are used.

- Overwrite: The conflict records in the destination instance are overwritten in case of a conflict.

  When the synchronization encounters a conflict of the preceding types, the conflict records in the destination instance are overwritten.

18.1.4.4.8.4. Synchronization restrictions

This section describes the restrictions in cross-cloud data synchronization using DTS.

Restrictions in data sources

Currently, only ApsaraDB RDS for MySQL instances support two-way synchronization. Other heterogeneous data sources do not support two-way synchronization.

The destination instance cannot be an RDS instance that runs in standard access mode and has only a public network address.

Restrictions in synchronization architecture

Currently, DTS only supports two-way synchronization between two ApsaraDB RDS for MySQL instances. Two-way synchronization between more than two instances is not supported.

Feature restrictions

- Incompatible with triggers

  When you synchronize an entire database and the database contains a trigger that updates the synchronization table, the synchronized data may be inconsistent.

  For example, the object to be synchronized is database A that contains table a and table b. Table a has a trigger that inserts a row to table b after the row is inserted to table a. In this case, if an INSERT operation is performed on table a in the source instance during synchronization, the data in table b is inconsistent between the source and destination instances.

  To resolve this problem, you must delete the trigger in the destination instance, so that the data in table b is only synchronized from the source instance.

- Restrictions in the RENAME TABLE operation

  The RENAME TABLE operation may result in inconsistent synchronization data. For example, if the object to be synchronized only includes table a and the **rename a to b** command is executed in the source instance during synchronization, subsequent operations to the renamed table b are not synchronized to the destination database. To solve this problem, you can synchronize the entire database where table a and table b are stored.

- Restrictions in DDL synchronization direction

To ensure the stability of a two-way synchronization channel, you can synchronize DDL updates on the same table in only one direction. For example, in A-to-B and B-to-A synchronization, you can implement DDL synchronization in either the **A-to-B** or **B-to-A** direction. If DDL synchronization is configured in one direction, it is not supported in the reverse direction.

18.1.4.4.8.5. Configure two-way data synchronization between ApsaraDB RDS for MySQL instances across regions

This topic describes how to configure two-way data synchronization between ApsaraDB RDS for MySQL instances across regions.

Prerequisites

The source and destination ApsaraDB RDS for MySQL instances are created.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Synchronization**.
3. On the **Synchronization Tasks** page, click **Create Synchronization Task** in the upper-right corner.

   > **Note**
   >
   > Source Instance Region: Select the region where the source RDS instance resides.
   >
   > Source Instance Type: Select the type of the source instance. In this example, select MySQL.
   >
   > Destination Instance Region: Select the region where the destination RDS instance resides.
   >
   > Destination Instance Type: Select the type of the destination instance. In this example, select MySQL.
   >
   > Synchronization Mode: Select the synchronization mode. In this example, select Two-Way Synchronization.
   >
   > Instances to Create: Set the number of instances that you want to create.

4. After you configure the preceding information, click **Create**.

   After you create a synchronization instance, go back to the **Synchronization Tasks** page. The new synchronization instance is in the Not Configured state and contains two synchronization tasks. You can configure two-way synchronization for the tasks.

5. Find one of the created synchronization tasks and click **Configure Synchronization Channel** in the Actions column.
6. Configure the parameters for the data synchronization task.
   - Synchronization Task Name

     We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
   - RDS instance ID

     You must specify the ID of the Apsara Stack tenant account to which the destination RDS instance belongs. You can then select an RDS instance ID from the Instance ID drop-down list.

     After you complete the preceding configurations, click **Set Whitelist and Next** to configure the RDS instance whitelists.
7. Configure the RDS instance whitelists.

In this step, DTS adds the CIDR blocks of DTS servers to the whitelists of the source and destination RDS instances. This ensures that DTS servers can connect to the RDS instances and the data synchronization task can be created.

We recommend that you do not remove the CIDR blocks of DTS servers from the whitelists of the RDS instances. This ensures the stability of the data synchronization task.

Click **Next** to create a data synchronization account.

8. Create a data synchronization account in the destination RDS instance.

   Create a data synchronization account named dtssyncwriter in the destination RDS instance. Do not delete the account during data synchronization. Otherwise, an interruption occurs.

9. Configure the synchronization policies and select the objects that you want to synchronize.

   After you create a data synchronization account, you must configure the synchronization policies and select the objects that you want to synchronize.

   - Exclude DDL Statements

     Specify whether to synchronize DDL statements in a specific direction. To include DDL statements, select **No**. To exclude DDL statements, select **Yes**. If you select No, DTS does not synchronize the DDL operations that are performed on a table in the opposite direction.

   - DML Statements for Synchronization

     Select the types of DML operations that you want to synchronize. By default, **Insert**, **Update**, and **Delete** are selected.

   - Conflict Resolution Policy

     Select the resolution policy for synchronization conflicts. By default, **TaskFailed** is selected.

     For example, if Node A is the primary business center and Node B is a secondary business center, you must give the priority to Node A. You must set the conflict resolution policy in the A-to-B direction to **Overwrite** and that in the B-to-A direction to **Ignore**.

   - Select Objects

     You can select databases and tables as the objects to be synchronized.

     If you select an entire database, all schema changes such as the CREATE TABLE and DROP VIEW operations that performed on the objects in the database are synchronized to the destination database.

     If you select a table, only the DROP TABLE, ALTER TABLE, TRUNCATE TABLE, RENAME TABLE, CREATE INDEX, and DROP INDEX operations that performed on this table are synchronized to the destination database.

10. Configure initial synchronization.

    Initial synchronization is the first step to start the synchronization task. During initial synchronization, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization.

    Initial synchronization includes **initial schema synchronization** and **initial full data synchronization**. You must select both **Initial Schema Synchronization** and **Initial Full Data Synchronization** in most cases.

    If the tables to be synchronized in one direction are also included in the objects to be synchronized in the opposite direction, DTS does not synchronize these tables during initial synchronization.

11. Run a precheck.

    After the data synchronization task is configured, DTS performs a precheck. Close the **Precheck** dialog box after the task passes the precheck.

    After the task is started, the task list appears. The task is in the **Performing Initial Sync** state. The duration of the initial synchronization depends on the data volume of the objects that you want to synchronize. After initial synchronization, the task status changes to **Synchronizing**. This indicates that the data synchronization task is created.

After the task is configured in one direction, the source and destination RDS instances of the task in the opposite direction cannot be changed.

12. Repeat steps 5 to 11 to configure the data synchronization task in the opposite direction.

18.1.4.5. Manage data synchronization instances

18.1.4.5.1. Specify the name of an object in the destination instance

After an object, such as a database or table, is synchronized from the source instance to the destination instance, the name of the object remains unchanged. You can use the object name mapping feature provided by DTS to specify a different name for the object in the destination instance.

Notes

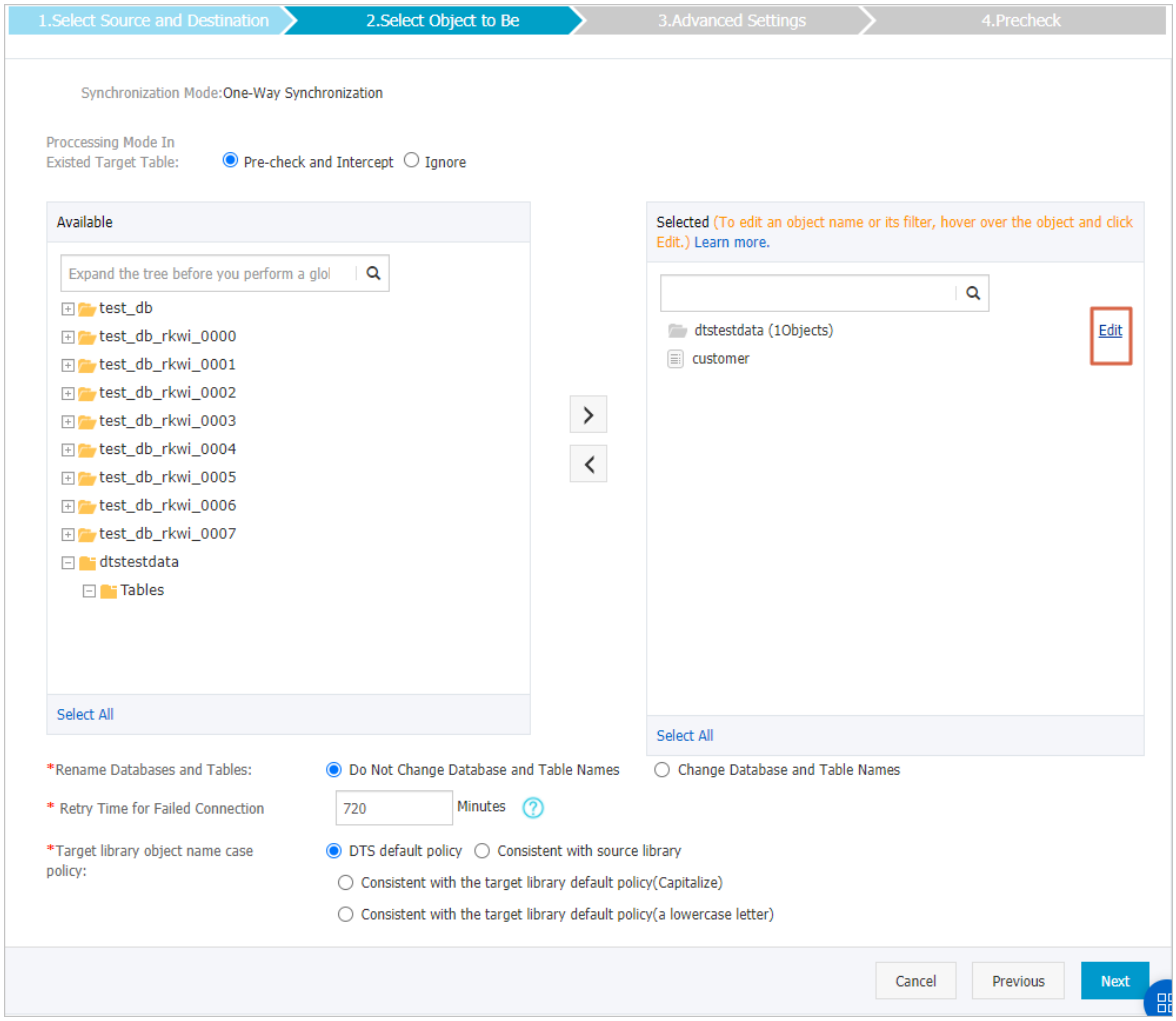
You can perform this operation only when a data synchronization task is configured and the current process is **Select Objects to Synchronize**.

> **Note** Do not perform this operation after the data synchronization task is started. Otherwise, the synchronization may fail.

Procedure

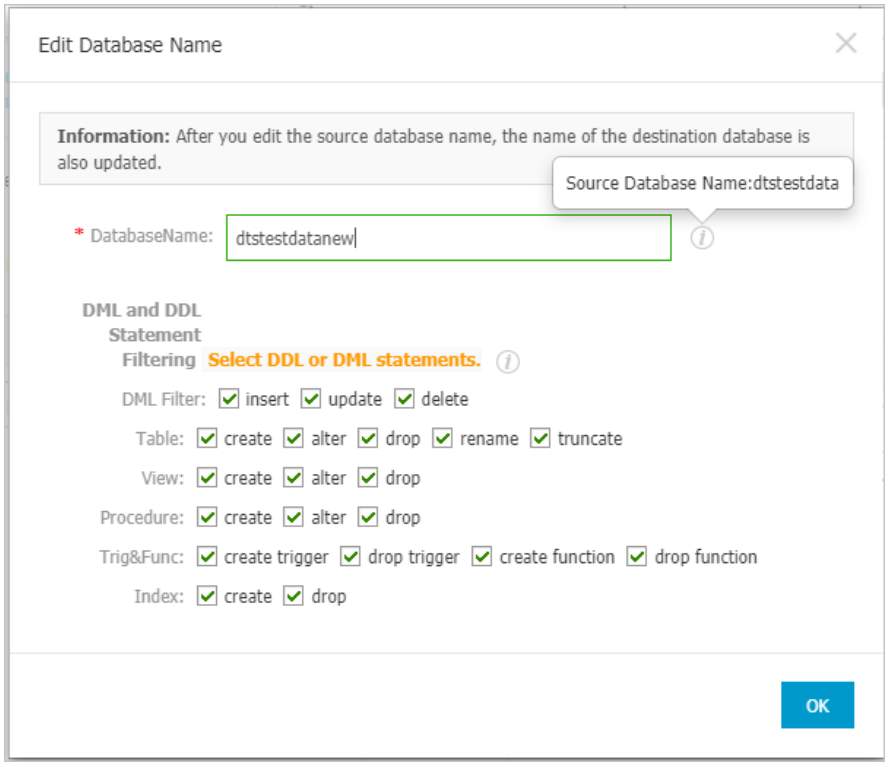
1. On the **Select Objects to Synchronize** page, move the required objects to the **Selected** section, move the pointer over a database or table, and then click **Edit**.

   > **Note** Different database types support different objects. If **Edit** appears when you move the pointer over the target object, the operation is supported.



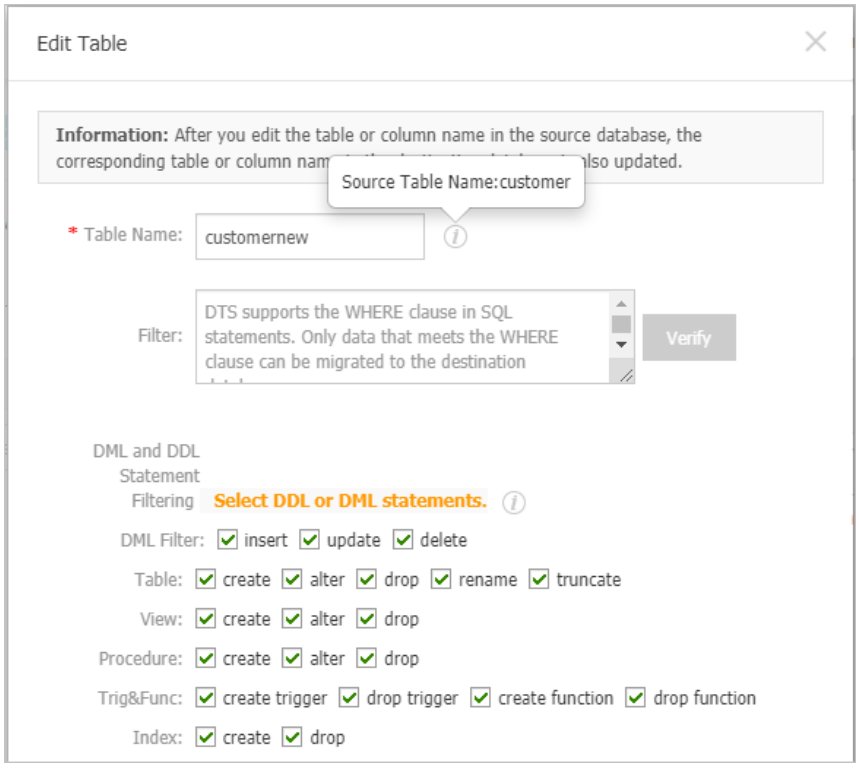
2. In the dialog box that appears, specify a name for the object in the destination instance.
   - Database name mapping

     In the **Edit Database Name** dialog box that appears, enter the database name that you want to use in the destination instance.



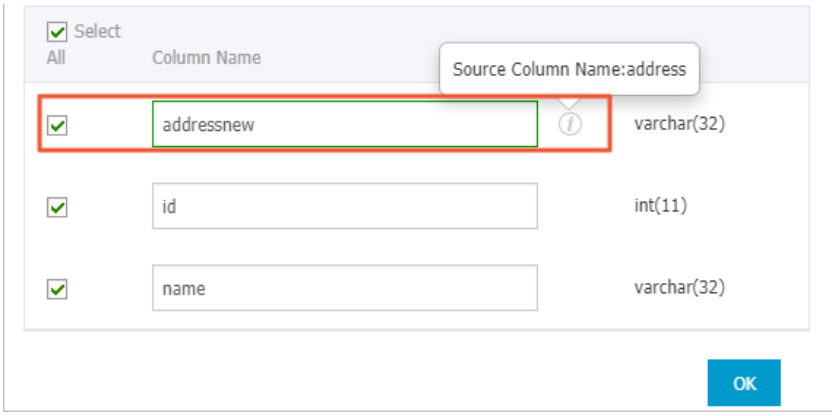
- Table name mapping

  In the **Edit Table** dialog box that appears, enter the table name that you want to use in the destination instance.



- Column name mapping

  In the **Edit Table** dialog box that appears, enter a new name for each column.



Note In this step, you can deselect columns that do not need to be synchronized.

3. Click **OK**.
4. Configure other parameters that are required for the data synchronization task.

18.1.4.5.2. Check the synchronization performance

DTS provides the trend charts of data synchronization tasks based on three performance metrics: bandwidth, synchronization speed (TPS), and synchronization delay. You can view the running status of data synchronization tasks in the DTS console.

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Synchronization**.
3. On the Synchronization Tasks page, click the ID of the data synchronization task that you want to check.

   The task details page appears.
4. On the task details page, click **Synchronization Performance** in the left-side navigation pane.
5. View the trend charts of synchronization performance.

   DTS provides the trend charts of data synchronization tasks based on three performance metrics: bandwidth, synchronization speed (TPS), and synchronization delay.

   - Bandwidth: the bandwidth of data that the data writing module pulls from the data pulling module per second. Unit: MB/s.
   - Synchronization speed (TPS): the number of transactions that DTS synchronizes to the destination instance per second.
   - Synchronization delay: the difference between the timestamp of the latest synchronized data in the destination instance and the current timestamp in the source instance. Unit: milliseconds.

18.1.4.5.3. Add objects to a data synchronization task

When a data synchronization task is running, you can add objects to the task or remove objects from the task. This topic describes how to add objects to a data synchronization task in the DTS console.

Limits

You can modify the required objects only when the data synchronization task is in the **Synchronizing** or **Synchronization Failed** state.

Start time of data synchronization

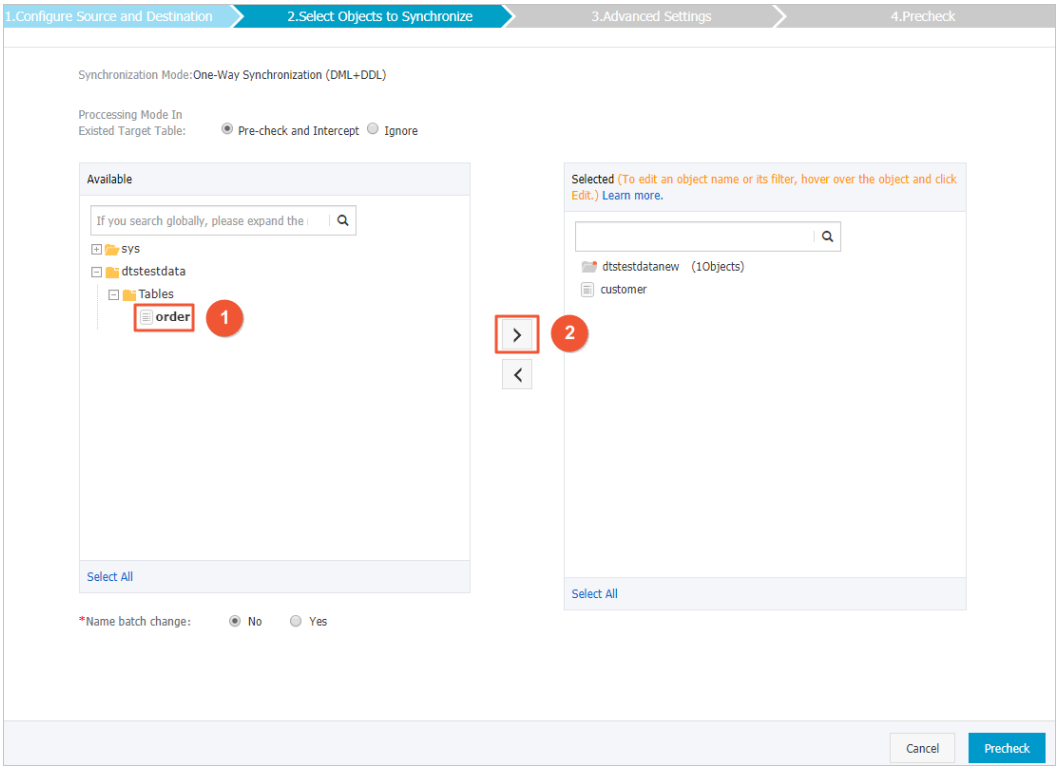
The time when DTS synchronizes data of new objects depends on whether initial synchronization is specified for the data synchronization task.

- If initial synchronization is specified, DTS synchronizes schemas and historical data, and then synchronizes incremental data.
- If initial synchronization is not specified, DTS synchronizes data after incremental data is generated on the source instance.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Synchronization**.
3. Find the data synchronization task and choose **More > Modify Objects to Synchronize** in the Actions column.
4. On the **Select Objects to Synchronize** tab, add objects based on your needs, as shown in Add objects to a data synchronization task.

   Add objects to a data synchronization task



5. Click **Precheck**.

   After the task passes the precheck, the objects are added to the data synchronization task.

   After the objects are added, if initial synchronization is specified for the data synchronization task, the task status changes from **Synchronizing** to **Synchronizing (The initial synchronization of the new objects is being performed.)**.

> **Note** You can click **View More** to view the initial synchronization progress of the new objects. After the initial synchronization on the new objects is complete, the task status returns to **Synchronizing**.

18.1.4.5.4. Remove objects from a data synchronization task

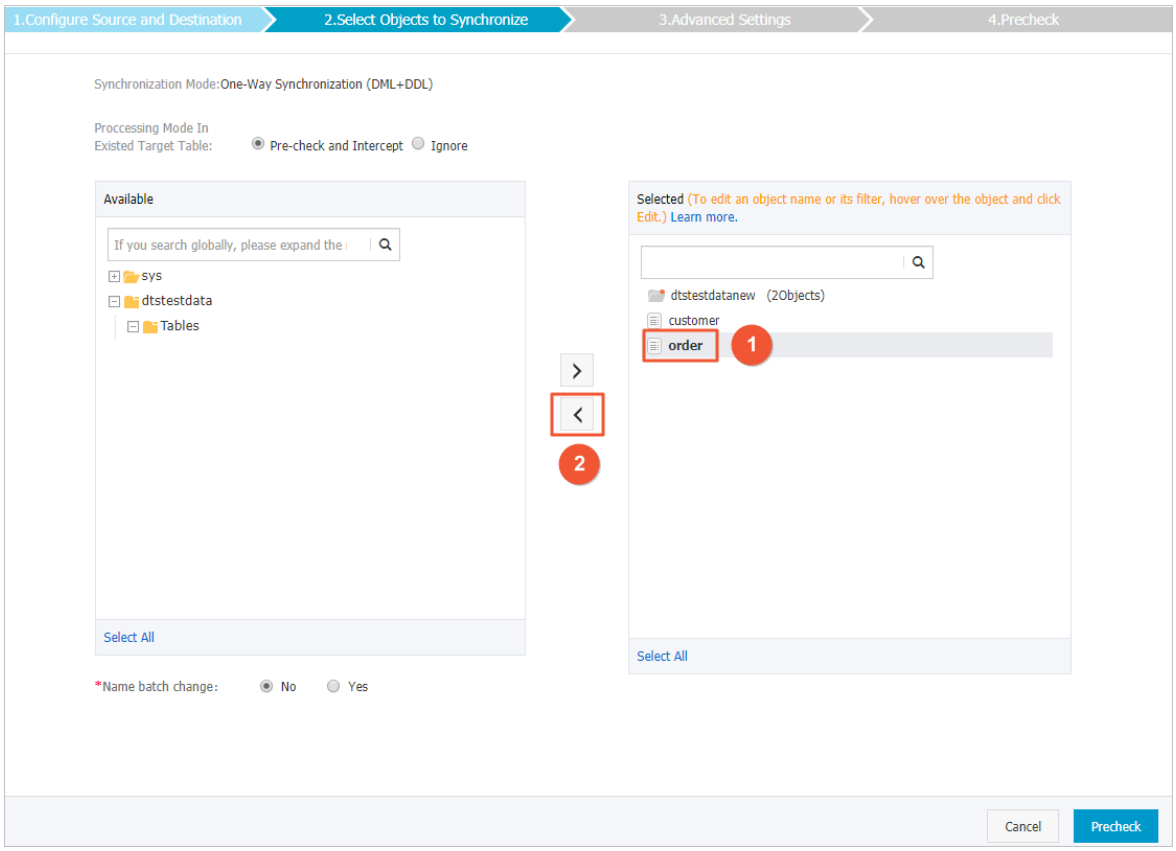
When a data synchronization task is running, you can add objects to the task or remove objects from the task. This topic describes how to remove objects from a data synchronization task in the DTS console.

Limits

You can modify the required objects only when the data synchronization task is in the **Synchronizing** or **Synchronization Failed** state.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Synchronization**.
3. Find the data synchronization task and choose **More > Modify Objects to Synchronize** in the Actions column.
4. On the **Select Objects to Synchronize** tab, remove objects based on your needs.



5. Click **Precheck** to run a precheck.

18.1.4.5.5. Troubleshoot precheck failures

Before Data Transmission Service (DTS) runs a data synchronization task, DTS performs a precheck. This topic describes the precheck items and how to troubleshoot precheck failures.

Source database connectivity

- Description

  DTS checks whether DTS servers can connect to the source RDS instance. DTS creates a connection to the source RDS instance by using the JDBC protocol. If the connection fails, the task fails to pass the precheck.

- Cause of failure
  - DTS does not support data synchronization between RDS instances in the region where the source instance resides.
  - The database account or password of the source instance is invalid.

- Solution

  Submit a ticket and contact Alibaba Cloud technical support.

Destination database connectivity

- Description

  DTS checks whether DTS servers can connect to the destination RDS instance. DTS creates a connection to the destination RDS instance by using the JDBC protocol. If the connection fails, the task fails to pass the precheck.

- Cause of failure
  - DTS does not support data synchronization between RDS instances in the region where the destination instance resides.
  - The database account or password of the destination instance is invalid.
- Solution

  Submit a ticket and contact Alibaba Cloud technical support.

Source database version

- Description

  DTS checks whether:

  i. The database version of the source RDS instance is supported by the data synchronization feature.

  ii. The database version of the destination RDS instance is the same as the database version of the source RDS instance.

- Cause of failure
  - The database version of the source RDS instance is earlier than the supported database versions. The data synchronization feature supports the following database versions: MySQL 5.1, 5.5, 5.6, and 5.7.
  - The database version of the destination RDS instance is earlier than the database version of the source RDS instance.
- Solution
  - If the database version of the source RDS instance is earlier than the supported database versions, upgrade the source RDS instance to MySQL 5.6 or 5.7 in the RDS console. Then, create a data synchronization task again.
  - If the database version of the destination RDS instance is earlier than the database version of the source RDS instance, upgrade the destination RDS instance to MySQL 5.6 or 5.7 in the RDS console. Then, create a data synchronization task again.

Database existence

DTS checks whether the destination database already exists in the destination instance. If the destination database does not exist in the destination instance, DTS automatically creates a database. However, DTS fails to create the database and reports a failure under the following circumstances:

- The database name contains characters other than lowercase letters, digits, underscores (_), and hyphens (-).
- The character set of the database is not UTF-8, GBK, Latin1, or UTF-8MB4.
- The account of the destination database does not have the read/write permissions on the source database.

  If the data source is an RDS instance, the task passes the precheck.

Source database permissions

DTS checks whether the account of the source database has the required permissions. If the account does not have the required permissions, the task fails to pass the precheck. If the source database is an RDS instance, the task passes the precheck.

Destination database permissions

- Description

  DTS checks whether the account of the destination database has the required permissions. If the account does not have the required permissions, the task fails to pass the precheck.

- Cause of failure
  - DTS fails to create a database account in the destination RDS instance.
  - DTS fails to grant the read/write permissions to the database account of the destination RDS instance.

- Solution

  Submit a ticket and contact Alibaba Cloud technical support.

Object name conflict

- Description

  DTS checks object names only if you select initial synchronization for a data synchronization task. DTS checks whether an object that you want to synchronize has the same name as an object in the destination RDS instance.

- Cause of failure

  If an object in the destination RDS instance has the same name as the object that you want to synchronize, the task fails to pass the precheck.

- Solution
  - Remove the conflicting object from the destination database.
  - Then, create a data synchronization task again. Select both Initial Schema Synchronization and Initial Full Data Synchronization.

Value of server_id in the source database

DTS checks whether the value of the server_id parameter in the source database is set to an integer that is greater than or equal to 2. If the data source is an RDS instance, the task passes the precheck.

Whether binary logging is enabled for the source database

DTS checks whether the binary logging feature is enabled for the source database. If the binary logging feature is disabled for the source database, the task fails to pass the precheck. If the data source is an RDS instance, the task passes the precheck.

Binary log format of the source database

DTS checks whether the binary log format of the source database is set to ROW. If the binary log format of the source database is not set to ROW, the task fails to pass the precheck. If the data source is an RDS instance, the task passes the precheck.

Integrity of the FOREIGN KEY constraints

- Description

  DTS checks whether the parent tables and child tables that have referential relationships with each other are all included in the required objects. The precheck allows DTS to protect the integrity of the FOREIGN KEY constraints.

- Cause of failure

  One or more child tables are included in the required objects. However, the parent tables that are referenced by the child tables are not included in the required objects. This impairs the integrity of the FOREIGN KEY constraints.

- Solution

  The following solutions are available:

  - Create a data synchronization task again and do not synchronize the child tables that fail to pass the precheck.
  - Create a data synchronization task again and add the parent tables to the required objects.
  - Remove the FOREIGN KEY constraints from the child tables that fail to pass the precheck. Then, create a data synchronization task again.

Storage engine

- Description

  DTS checks whether the required objects use the storage engines that are not supported by the data synchronization feature, such as FEDERATED, MRG_MyISAM, and TokuDB.

- Cause of failure

  If the storage engine of a source table is FEDERATED, MRG_MyISAM, or TokuDB, the task fails to pass the precheck.

- Solution

  Change the unsupported storage engine to InnoDB and create a data synchronization task again.

Character set

- Description

  DTS checks whether the required objects use the character sets that are not supported by the data synchronization feature, such as the UCS-2 character set.

- Cause of failure

  If the character sets used by the required objects are not supported by the data synchronization feature, the task fails to pass the precheck.

- Solution

  Change the unsupported character sets to UTF-8, GBK, or Latin1. Then, create a data synchronization task again.

Complicated topologies

- Description

  DTS checks whether the topology that you specify for the source and destination RDS instances is supported.

- Cause of failure
  - The source RDS instance in the current task is being used as the destination instance of another task.
  - The destination RDS instance in the current task is being used as the source or destination instance of another task.
  - The objects that you want to synchronize in the current task are being synchronized by an existing task. The two tasks have the same source and destination RDS instances.

- Solution
  - If the task that you want to create has the same source and destination RDS instances as an existing task, you can add the required objects to the existing task. You do not need to create another task to synchronize these objects.
  - If the task that you want to create conflicts with an existing task, wait until the existing task is completed before you create a data synchronization task again.

Format of the MySQL database password

DTS checks whether the format of the password that is used to access the source database is no longer valid. If the data source is an RDS instance, the task passes the precheck.

18.1.5. Change tracking

18.1.5.1. Overview

You can use Data Transmission Service (DTS) to track data changes from ApsaraDB RDS for MySQL instances in real time. This feature applies to the following scenarios: lightweight cache updates, business decoupling, asynchronous data processing, and synchronization of extract, transform, and load (ETL) operations.

Supported databases

- User-created MySQL databases or ApsaraDB RDS for MySQL
- PolarDB-X (formerly known as DRDS)
- User-created Oracle database

Objects for change tracking

The objects for change tracking include tables and databases.

In change tracking, data changes include data manipulation language (DML) operations and data definition language (DDL) operations. When you configure change tracking, you must select operation types.

Change tracking tasks

A change tracking task is the basic unit of change tracking and data consumption. To track data changes from an RDS instance, you must create a change tracking task in the DTS console for the RDS instance. The change tracking task pulls data changes from the RDS instance in real time and locally stores the data changes. You can use the DTS SDK to consume the tracked data. You can also create, manage, or delete change tracking tasks in the DTS console.

18.1.5.2. Create a change tracking instance

Before you configure a task to track data changes, you must create a change tracking instance. This topic describes how to create a change tracking instance in the Data Transmission Service (DTS) console.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Change Tracking**.
3. In the upper-right corner, click **Create Change Tracking Task**.
4. In the Create DTS Instances dialog box, select a region, and enter the number of change tracking instances that you want to create.

   > **Note** In the Create DTS Instances dialog box, you can view the total number of instances, the number of existing instances, and the number of instances that can be created.

5. Click **Create**.

18.1.5.3. Configure change tracking tasks

18.1.5.3.1. Track data changes from a user-created MySQL database or an ApsaraDB RDS for MySQL instance

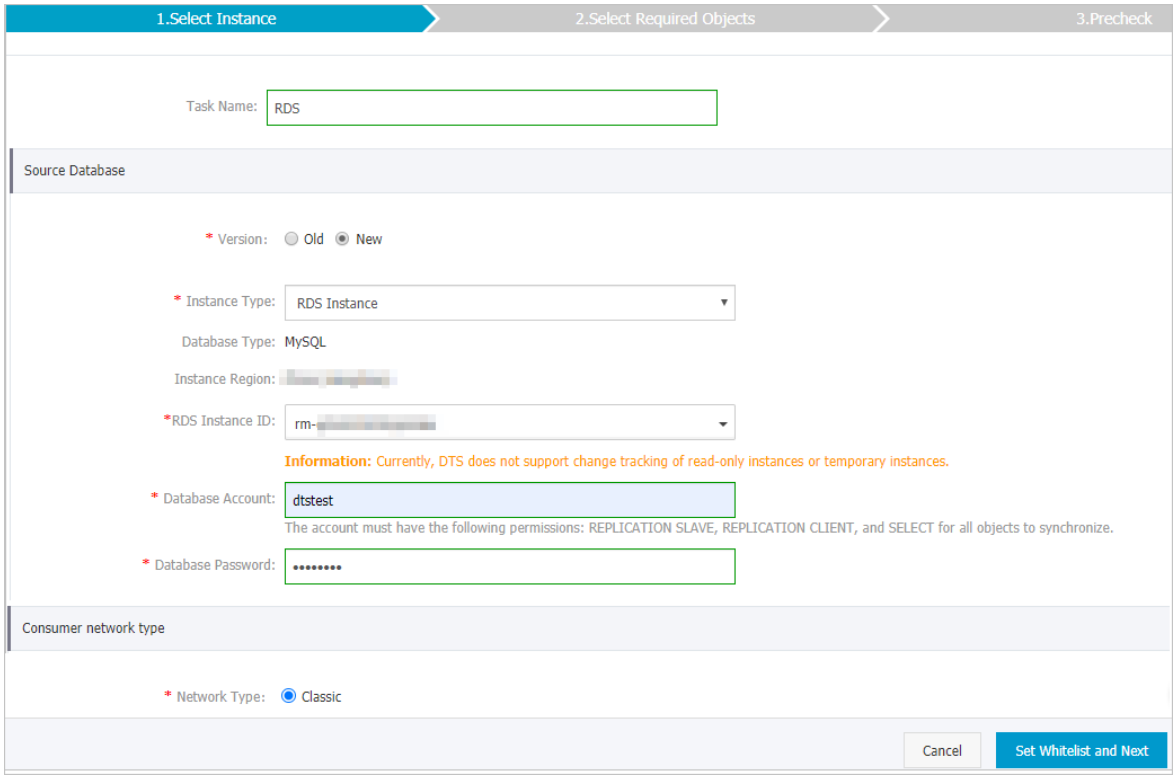
You can use Data Transmission Service (DTS) to track data changes in real time. This feature applies to the following scenarios: lightweight cache updates, business decoupling, asynchronous data processing, and synchronization of extract, transform, and load (ETL) operations. This topic describes how to track data changes from a user-created MySQL database or an ApsaraDB RDS for MySQL instance.

Prerequisites

The database version is 5.1, 5.5, 5.6, or 5.7.

Procedure

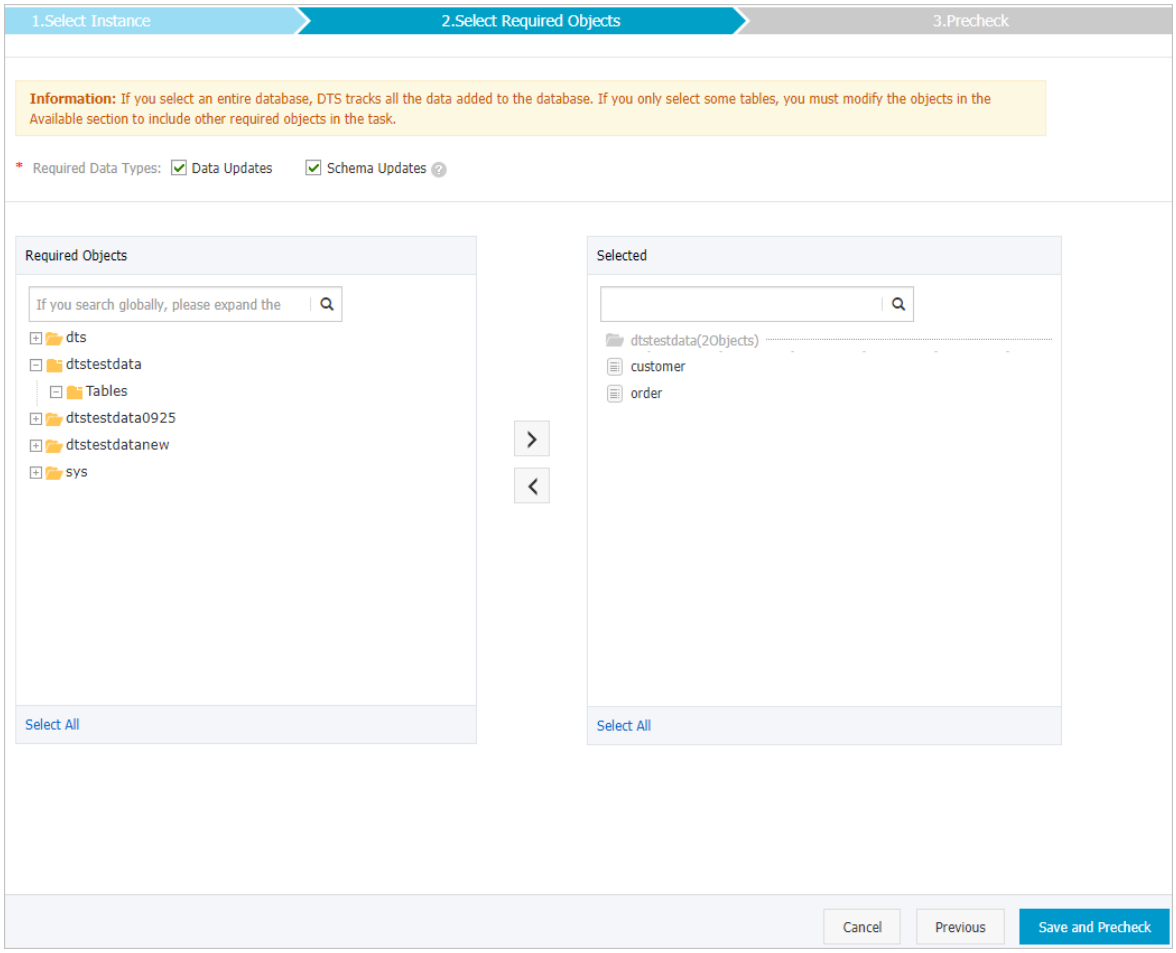
1.
2.
3. Configure the source database.



Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
	Version	Select a version based on your business requirements: ○ If the source database is a user-created MySQL database, select **Old**. ○ If the source database is ApsaraDB RDS for MySQL, select **New**. **Note** You can follow the same procedure to configure a change tracking task when you select Old or New. In this example, select New.
	Instance Type	Select **RDS Instance**.
	Database Type	This parameter is set to **MySQL** and cannot be changed.

Section	Parameter	Description
Source Database	Instance Region	The source region that you selected when you created the change tracking instance. You cannot change the value of this parameter.
	RDS Instance ID	Select the ID of the RDS instance from which you want to track data changes. **Note** A read-only instance or temporary instance cannot be used as the source instance for change tracking.
	Database Account	Enter the database account of the source RDS instance. **Note** • The account must have the SELECT permission on the required objects, the REPLICATION SLAVE permission, and the REPLICATION CLIENT permission. • If the database engine of the source RDS instance is **MySQL 5.5** or **MySQL 5.6**, you do not need to configure the **database account** or **database password**.
	Database Password	Enter the password of the source database account.
Consumer Network Type	Network Type	**Classic** is selected by default. **Note** • This parameter is available only if the Version parameter is set to **New**. • If you track data changes over internal networks, the network latency is minimal.

4.
5. Select the data change types and objects.



6.
7.

What's next

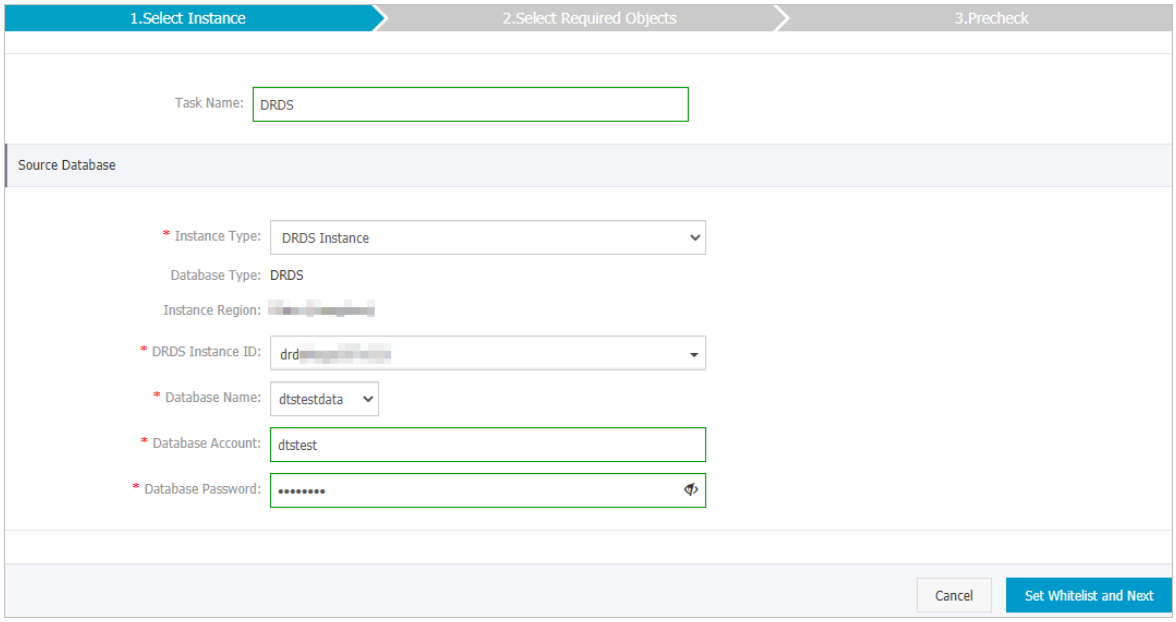
- Previous change tracking feature: Run the SDK demo code
- New change tracking feature: Use a Kafka client to consume tracked data

18.1.5.3.2. Track data changes from a PolarDB-X instance

You can use Data Transmission Service (DTS) to track data changes in real time. This feature applies to the following scenarios: lightweight cache updates, business decoupling, asynchronous data processing, and synchronization of extract, transform, and load (ETL) operations. This topic describes how to track data changes from a PolarDB-X instance. PolarDB-X is formerly known as Distributed Relational Database Service (DRDS).

Procedure

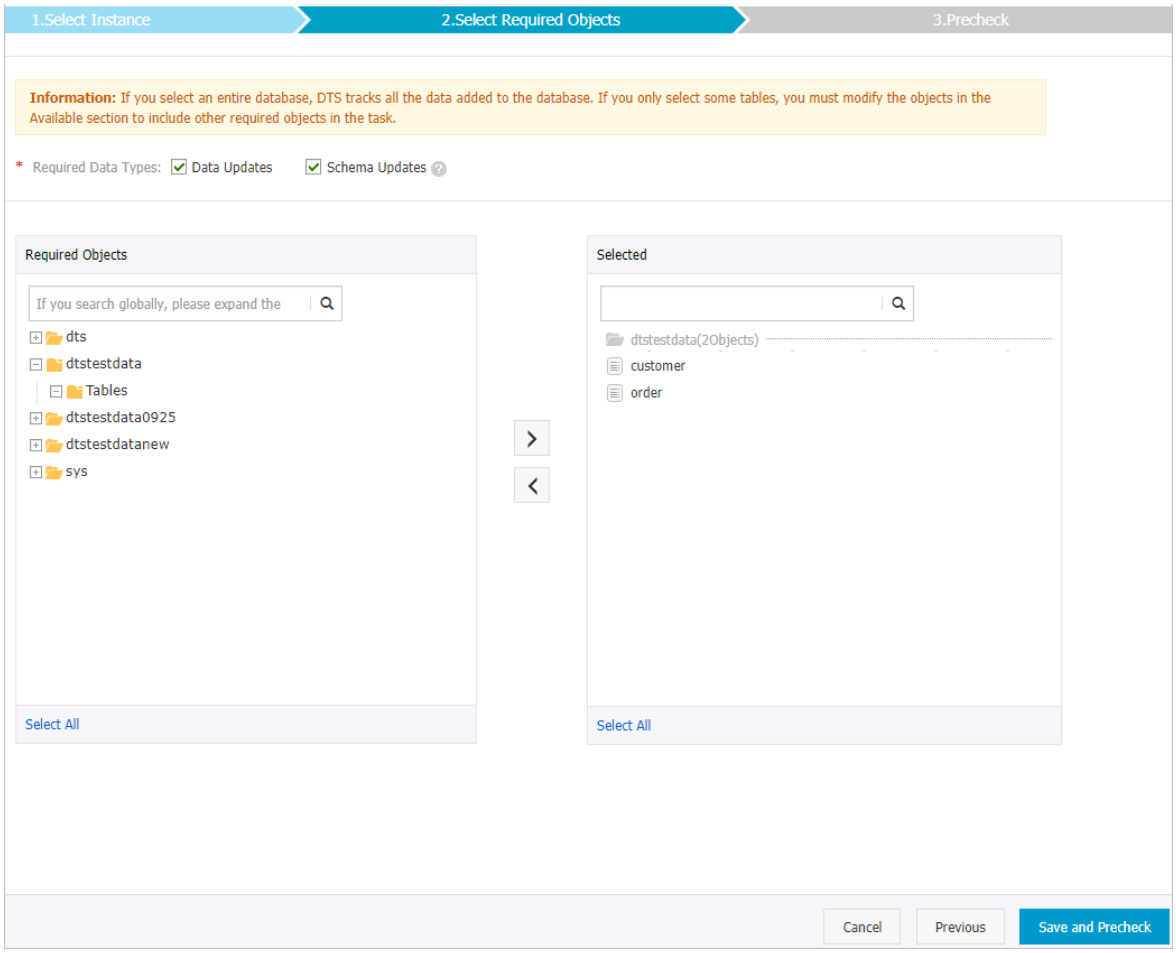
1.
2.
3. Configure the source database.



Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	This parameter is set to **DRDS Instance** and cannot be changed.
	Database Type	This parameter is set to **DRDS** and cannot be changed.
	Instance Region	The region of the source instance. The region is the same as the region that you selected when you created the change tracking task. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the PolarDB-X instance.
	Database Name	Select the ID of the source database in the PolarDB-X instance.
	Database Account	Enter the database account of the PolarDB-X instance.
	Database Password	Enter the password of the source database account.

4.

5. Select the data change types and objects.



6.
7.

18.1.5.3.3. Track data changes from a user-created Oracle database

You can use Data Transmission Service (DTS) to track data changes in real time. This feature applies to the following scenarios: lightweight cache updates, business decoupling, and synchronization of extract, transform, and load (ETL) operations. This topic describes how to track data changes from a user-created Oracle database.

Prerequisites

- The version of the user-created Oracle database is 9i, 10g, 11g, 12c, 18c, or 19c.
- Supplemental logging, including SUPPLEMENTAL_LOG_DATA_PK and SUPPLEMENTAL_LOG_DATA_UI, is enabled for the user-created Oracle database. For more information, see Supplemental Logging.
- The user-created Oracle database is running in ARCHIVELOG mode. Archived log files are accessible and a suitable retention period is set for archived log files. For more information, see Managing Archived Redo Log Files.

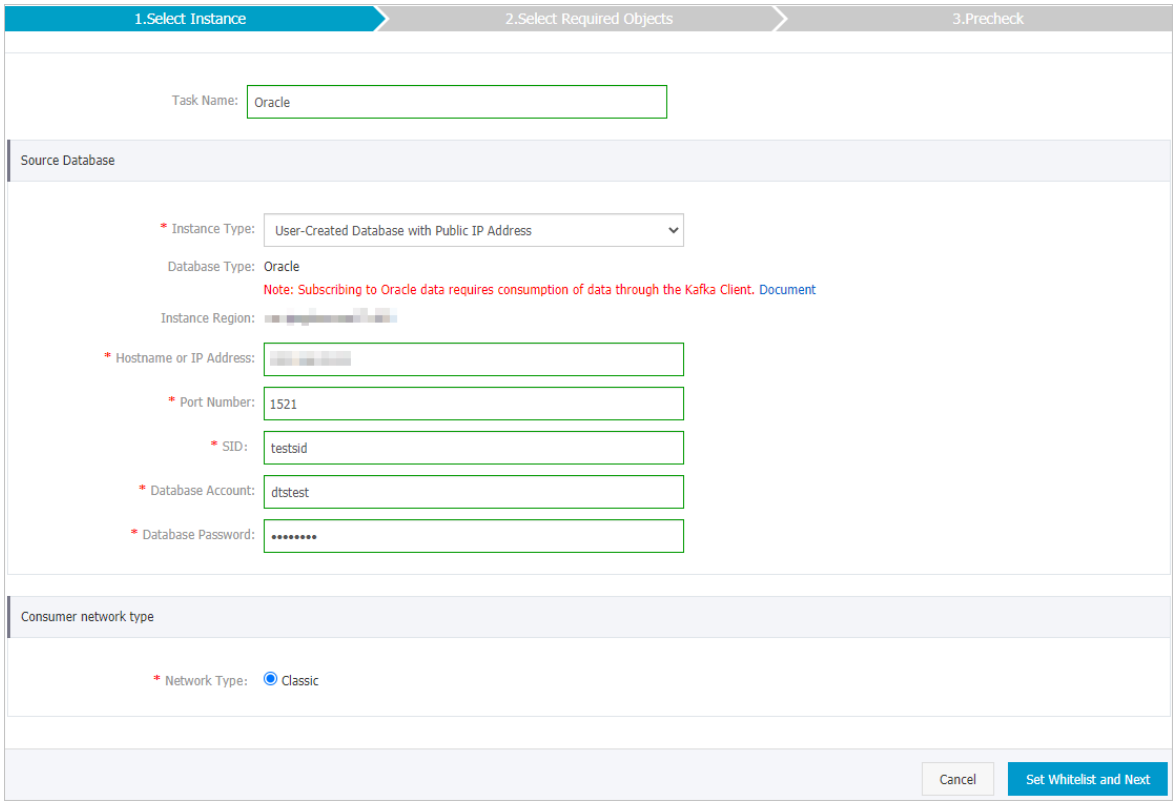
Precautions

- DTS does not track data definition language (DDL) operations that are performed by gh-ost or pt-online-schema-change. Therefore, the change tracking client may fail to write the consumed data to the destination tables due to schema conflict.

- If the source database is used in another task, for example, it is used in a running data migration task, DTS may track data changes of other objects. In this case, you must use the change tracking client to filter the tracked data.
- If you perform a primary/secondary switchover on the source database when the change tracking task is running, the task fails.

Procedure

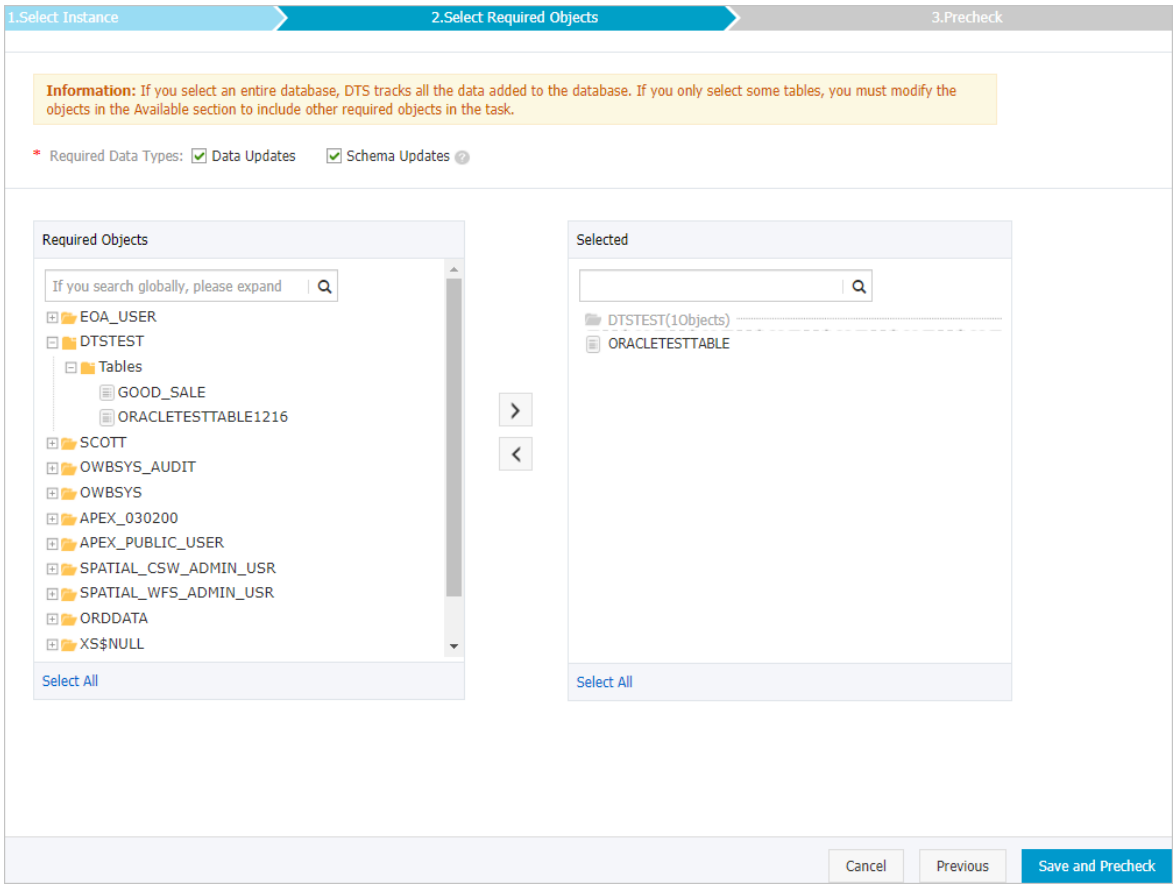
1. Create a change tracking instance.
2. Find the change tracking instance that you created, and click **Configure Channel** in the **Actions** column.
3. Configure the source database and network type.



Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select an instance type based on the deployment of the source database. In this example, select **User-Created Database with Public IP Address**.
	Database Type	This parameter is set to Oracle and cannot be changed.
	Instance Region	The source region that you selected when you created the change tracking instance. You cannot change the value of this parameter.
	Hostname or IP Address	Enter the hostname or IP address of the user-created Oracle database.
	Port Number	Enter the service port number of the user-created Oracle database.
	SID	Enter the system ID (SID) of the user-created Oracle database.

Section	Parameter	Description
	Database Account	Enter the account of the user-created Oracle database. **Note** The account must have the database administrator (DBA) permission.
	Database Password	Enter the password of the source database account.
Consumer Network Type	N/A	**Classic** is selected by default. **Note** If you track data changes over internal networks, the network latency is minimal.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Select the data change types and objects.



Parameter	Description

Parameter	Description
Required Data Types	◦ **Data Updates** DTS tracks data updates of the selected objects, including the INSERT, DELETE, and UPDATE operations. ◦ **Schema Updates** DTS tracks the create, delete, and modify operations that are performed on all object schemas of the source instance. You must use the change tracking client to filter the required data. **Note** ◦ If you select a database as the object, DTS tracks data changes of all objects, including new objects in the database. ◦ If you select a table as the object, DTS only tracks data changes of this table. In this case, if you want to track data changes of another table, you must add the table to the required objects. For more information, see Modify the objects for change tracking.
Required Objects	Select tables or databases from the **Required Objects** section and click the › icon to move them to the **Selected** section.

6. In the lower-right corner of the page, click **Save and Precheck**.

   Note You can start a change tracking task only after the task passes the precheck. If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.** After the change tracking task is configured, DTS performs initial change tracking, which takes about 1 minute. After the initial change tracking is complete, you can consume the tracked data.

What's next

Use a Kafka client to consume tracked data

18.1.5.4. Manage change tracking tasks

18.1.5.4.1. Modify the consumption checkpoint

During data consumption, you can modify the consumption checkpoint of a change tracking task based on your business requirements. After you modify the consumption checkpoint, the downstream SDK client will consume the data that is generated after the specified time.

Prerequisites

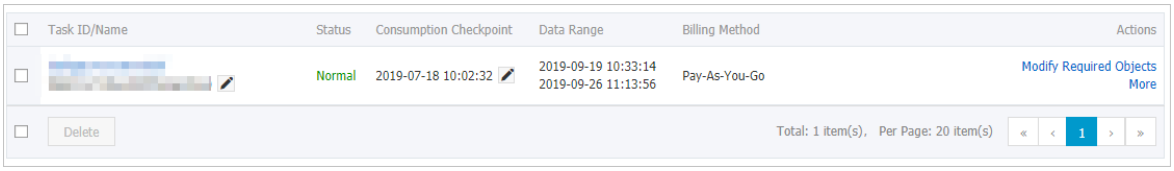
A change tracking task is created. For more information, see Track data changes from a user-created MySQL database or an ApsaraDB RDS for MySQL instance (previous version) or Track data changes from a PolarDB-X instance.

Procedure

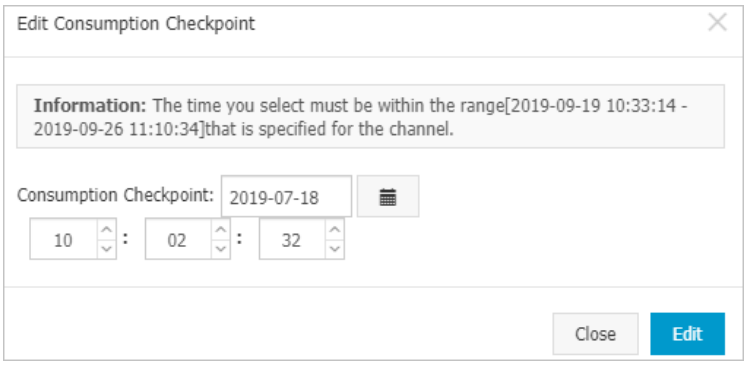
1. Stop all downstream SDK consumption processes.

> **Note** We recommend that you perform this operation during off-peak hours to avoid service interruption.

2. Log on to the DTS console.
3. In the left-side navigation pane, click **Change Tracking**.
4. Find the change tracking task, move the pointer over the **Consumption Checkpoint** column, and then click the icon.



5. In the Modify Consumption Checkpoint dialog box, specify a new consumption checkpoint.



> **Note** The selected time range must be within the time range of the tracked data. For more information, see the prompt in the dialog box.

6. Click **Modify**.
7. Restart the downstream SDK consumption processes.
   The downstream SDK client tracks data changes from the new consumption checkpoint.

18.1.5.4.2. Modify the objects for change tracking

DTS allows you to add or remove the objects for change tracking in the consumption process. This topic describes how to modify the objects for change tracking.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Change Tracking**.
3. Find the change tracking task, and click **Modify Required Objects** in the **Actions** column.
4. In the **Select Required Objects** step, add or remove the objects for change tracking.
   - Add the objects for change tracking

     In the **Required Objects** section, select one or more objects and click the › icon to add the objects to the **Selected** section.
   - Remove the objects for change tracking

In the **Selected** section, select one or more objects and click the icon to move the objects to the **Required Objects** section.

5. In the lower-right corner of the page, click **Save and Precheck**.

18.1.5.4.3. Create a consumer group

You can manage consumer groups of a change tracking task in the DTS console. This topic describes how to create a consumer group.

Prerequisites

A change tracking task is created. For more information, see Track data changes from a user-created MySQL database or an ApsaraDB RDS for MySQL instance (new version) or Track data changes from a PolarDB-X instance.

Note

- You can create multiple consumer groups (up to 20) in a change tracking instance to repeatedly consume data.
- A consumer group consumes each message only once, and only one consumer can consume data.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Change Tracking**.
3. Find the change tracking task and click the task ID.
4. In the left-side navigation pane, click **Consume Data**.
5. On the **Consume Data** page, click **Add Consumer Group** in the upper-right corner.
6. In the dialog box that appears, set the parameters for the consumer group.

Parameter	Description
Consumer Group Name	Enter a new name for the consumer group. We recommend that you use an informative name for easy identification.
Username	Enter the username of the consumer group. ○ A username must contain one or more of the following character types: uppercase letters, lowercase letters, digits, and underscores (_). ○ The username must be 1 to 16 characters in length.
Password	Enter the password that corresponds to the username of the consumer group. ○ A password must contain two or more of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ The password must be 8 to 32 characters in length.
Confirm Password	Enter the new password again.

7. Click **Create**.

18.1.5.4.4. Manage consumer groups

You can manage consumer groups of a change tracking task in the DTS console. This topic describes how to modify the password of a consumer group and how to delete a consumer group.

Prerequisites

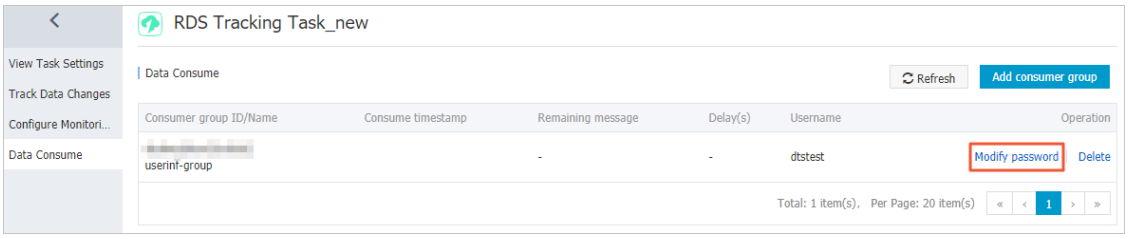
Create a consumer group

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Change Tracking**.
3. Find the change tracking task and click the task ID.
4. In the left-side navigation pane, click **Consume Data**.
5. Modify the password of a consumer group or delete a consumer group.

   Modify the password of a consumer group

   i. On the **Consume Data** page, find the target consumer group and click **Modify Password** in the **Actions** column.

   ii. In the **Modify Password** dialog box that appears, enter the **old password** and **new password**, and enter the new password again in the **Confirm Password** field.

   > Note
   > - A password must contain two or more of the following character types: uppercase letters, lowercase letters, digits, and special characters.
   > - The password must be 8 to 32 characters in length.

   iii. Click **Modify**.

   Delete a consumer group

   > Note After a consumer group is deleted, the data in the group will be cleared and cannot be recovered. We recommend that you use caution when performing this operation.

   i. On the **Consume Data** page, find the target consumer group and click **Delete** in the **Actions** column.

   ii. In the **Delete Consumer Group** message that appears, click **OK**.

18.1.5.5. Use the SDK to consume tracked data

18.1.5.5.1. Methods provided by SDK

You can use the SDK demo code that is provided by DTS to consume tracked data. This topic describes the methods that are available for the SDK classes.

Methods of the RegionContex class

Method	Description

Method	Description
`setAccessKey(accessKey)`	Specifies the AccessKey ID of the Alibaba Cloud account to which the source instance belongs.
`setSecret(AccessKeySecret)`	Specifies the AccessKey secret of the Alibaba Cloud account to which the source instance belongs.
`setUsePublicIp(usePublicIp)`	Specifies whether to track data changes over the Internet. **Note** DTS can track data changes only over the Internet. Therefore, set the usePublicIp parameter to `true`.
`context.setUseBinary(boolean useBinary)`	Specifies whether to enable the binary packaging feature. Valid values: True and False. We recommend that you enable this feature to improve consumption performance.
`context.setUseDrcNet(boolean useDrcNet)`	Specifies whether to enable the network optimization feature. Valid values: True and False. We recommend that you enable this feature to improve consumption performance.

Methods of the ClusterClient class

Method	Description
`void addConcurrentListener(ClusterListener arg0)`	Adds a downstream listener to retrieve data changes from a change tracking instance. **Note** The `ClusterListener arg0` parameter specifies an object of the `ClusterListener` class.
`void askForGUID(String arg0)`	Retrieves data changes from a change tracking instance. Set the String arg0 parameter to the ID of the change tracking instance.
`List<ClusterListener> getConcurrentListeners()`	Queries the list of listeners in a ClusterClient object. The return type is `List <ClusterListener>`.
`void start()`	Starts the SDK client to start change tracking.
`void stop()`	Stops the SDK client to stop change tracking. **Note** Data pulling and notification callback are performed in the same thread of the SDK client. If the consumption code of the notify() method contains a function that prevents signal interruptions, the stop() function may fail to terminate the SDK client.

Methods of the ClusterListener class

The `void notify(List<ClusterMessage> arg0)` method specifies the consumption mode of tracked data. When the DTS SDK receives the data, it uses the notify() function to notify a ClusterListener object to consume the data. Then, the SDK displays the data on the screen.

Methods of the ClusterMessage class

> **Note** Each ClusterMessage object stores the data record of a transaction. Each data record in the transaction is stored by using a Record object.

Method	Description
`Record getRecord()`	Retrieves a change record from a ClusterMessage object. The change record contains an entry in the binary log file, such as a BEGIN, COMMIT, UPDATE, or INSERT operation.
`void ackAsConsumed`	After the data consumption is complete, you must call this method to send an ACK packet to instruct the DTS server to update the consumer offset. This ensures the integrity of the consumed data after an abnormal SDK client restarts. **Note** If a downstream SDK client restarts after a breakdown, the client resumes change tracking from the last consumer offset.

Methods of the Record class

The `String getAttribute(String key)` method retrieves the attribute values in a Record object. The following table describes the parameters that are available when you call this method.

Parameter	Description
`record_id`	The ID of the record. **Note** The record ID may not increment during the change tracking process.
`instance`	The endpoint that is used to connect to the database instance. The format is <IP address>:<Port number>.
`source_type`	The engine type of the database instance. The value is set to MySQL.
`source_category`	The type of the record. The value is set to full_recorded.
`timestamp`	The binlog timestamp that is generated when the SQL statement is executed in the source database.

Parameter	Description
`checkpoint`	The checkpoint of the binary log file. The format is `binlog_offset@binlog_file`. **Note** The `binlog_offset` parameter indicates the offset of a record in the binary log file. The `binlog_file` parameter indicates the numerical suffix of the binary log file. For example, if the name of a binary log file is mysql-bin.0008, the value of the binlog_file parameter is 8.
`record_type`	The operation type. Valid values: insert, update, delete, replace, ddl, begin, commit, and heartbeat. **Note** A heartbeat record indicates the heartbeat table that is defined by DTS. The system generates one heartbeat record per second to detect whether the change tracking instance is running as expected.
`db`	The name of the database.
`table_name`	The name of the table.
`record_recording`	The encoding format.
`primary`	The name of the primary key column. If the primary key is a composite key, separate column names with commas (,).
`fields_enc`	The encoding of each field value. Separate fields with commas (,). **Note** If a field value is not of the character type, the encoding of this field value is null.

The following table lists the methods that are preset in the SDK demo code. You can call these methods to retrieve the attribute values in a Record object.

Method	Description
`Type getOpt()`	Queries the operation type.
`String getCheckpoint()`	Queries the checkpoint of the binary log file.
`String gettimestamp()`	Queries the timestamp of the binary log file.
`String getDbname()`	Queries the database name.
`String getTablename()`	Queries the table name.

Method	Description
`String getPrimaryKeys()`	Queries the name of the primary key column.
`DBType getDbType()`	Queries the database type.
`String getServerId()`	Queries the endpoint that is used to connect to the database instance.
`int getFieldCount()`	Queries the number of fields.
`List<Field> getFieldList()`	Queries the definitions of all fields, the pre-change image values, and the post-change image values. For more information, see Methods of the Field class.
`Boolean isFirstInLogevent()`	Checks whether the record is the first transaction log in a large volume of data changes. The return value is True or False.

Methods of the Field class

Method	Description
`String getEncoding()`	Obtains the encoding format of the field value.
`String getFieldname()`	Queries the name of the field.
`Type getType()`	Queries the data type of the field.
`ByteString getValue()`	Queries the value of the field. The return type is ByteString. If the field is not specified, the method returns `NULL`.
`Boolean isPrimary()`	Checks whether the field is a primary key column. The return value is True or False.

18.1.5.5.2. Quick start

This section describes how to use the DTS Java SDK to perform some basic operations.

Initialize a RegionContext object

A RegionContext object stores the settings of authentication credentials and network access mode. The following code shows how to initialize a RegionContext object.

```
import java.util.List;
import com.aliyun.drc.clusterclient.ClusterClient;
import com.aliyun.drc.clusterclient.DefaultClusterClient;
import com.aliyun.drc.clusterclient.RegionContext;
public class MainClass
{
      public static void main(String[] args) throws Exception {
        // Create a RegionContext object.
          RegionContext context = new RegionContext();
          context.setAccessKey("<AccessKey>");
          context.setSecret("<AccessKeySecret>");
          context.setUsePublicIp(true);
          // Create a ClusterClient object.
          final ClusterClient client = new DefaultClusterClient(context);
          // Other invocation code.
          ...
    }
}
```

Initialize a Listener object

Data consumption is implemented by using an object of the Listener class. After you initialize the ClusterClient object, you must add a Listener object. The Listener object uses the notify() method to receive and consume the tracked data. The following code shows how to display the tracked data on the screen.

```
import com.aliyun.drc.clusterclient.ClusterClient;
import com.aliyun.drc.clusterclient.ClusterListener;
import com.aliyun.drc.clusterclient.DefaultClusterClient;
import com.aliyun.drc.clusterclient.RegionContext;
import com.aliyun.drc.clusterclient.message.ClusterMessage;
public class MainClass
{
    public static void main(String[] args) throws Exception {
        // Initialize the RegionContext object.
        ...
        //Initialize the ClusterClient object.
        ...
        ClusterListener listener = new ClusterListener(){
             @Override
             public void notify(List<ClusterMessage> messages) throws Exception {
                  for (ClusterMessage message : messages) {
                    // Display the tracked data on the screen.
                      System.out.println(message.getRecord() + ":" + message.getRecord().getTablename
() + ":"
                      + message.getRecord().getOpt());
                      // Call the following method to send an ACK packet to the DTS server.
                      message.ackAsConsumed();
               }
      }
     }
}
```

DTS saves the consumption checkpoints of the SDK to the DTS server. This simplifies disaster recovery during the use of the SDK. The ackAsConsumed() method sends the checkpoint and timestamp of the latest data record that was consumed by the DTS SDK to the DTS server. If the SDK restarts due to an error, the SDK obtains the consumption checkpoint from the DTS server. The SDK resumes data consumption from the checkpoint. This ensures that the SDK does not consume duplicate data.

Start the ClusterClient object

Use the following code:

```
import java.util.List;
import com.aliyun.drc.clusterclient.ClusterClient;
import com.aliyun.drc.clusterclient.ClusterListener;
import com.aliyun.drc.clusterclient.DefaultClusterClient;
import com.aliyun.drc.clusterclient.RegionContext;
import com.aliyun.drc.clusterclient.message.ClusterMessage;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
public class MainClass
{
   public static void main(String[] args) throws Exception {
    // Initialize the RegionContext object.
    ...
    // Initialize the ClusterClient object.
    ...
    // Initialize the ClusterListener object.
    ...
    // Add a Listener class.
      client.addConcurrentListener(listener);
      // Specify the ID of the change tracking instance.
      client.askForGUID("dts_rdsrjiei2u2afnb_DSF");
      // Start a background thread. The main thread cannot exit.
      client.start();
}
```

The askForGUID() method sets the ID of the change tracking instance. You can obtain the ID of the change tracking instance from the DTS console. After the ID of the change tracking instance is specified in the askForGUID() method, the SDK retrieves incremental data from this instance.

Before you can start a ClusterClient object, you must add a Listener class to the ClusterClient object. When the ClusterClient object pulls incremental data from the change tracking instance, it also calls the notify() method of the Listener class to consume data.

18.1.5.5.3. Parse tracked SQL statements

You can use the DTS SDK to track data changes. DTS records the tracked data changes in a custom format. This topic describes how to parse various types of SQL statements.

Parse a DDL statement

If a data definition language (DDL) operation is performed in the source database, the operation type of the data record is DDL. The DDL statement is stored in the value of the first column. You can use the following sample code to parse the DDL statement:

```
String ddl_string;
Record.Type type=record.getOpt();
if(type.equals(Record.Type.DDL)){
    List<DataMessage.Record.Field> fields = record.getFieldList();
    ddl_string = fields.get(0).getValue().toString();
}
```

Parse an INSERT statement

If an INSERT operation is performed in the source database, the operation type of the data record is INSERT. You can use the following sample code to parse the INSERT statement:

```
StringBuilder insert_string=new StringBuilder();
Record.Type type=record.getOpt();
DataMessage.Record.Field field;
StringBuilder FieldName=new StringBuilder();
StringBuilder FieldValue = new StringBuilder();
if(type.equals(Record.Type.INSERT)){
       int i=0;
       List<DataMessage.Record.Field> fields = record.getFieldList();
       for (; i < fields.size(); i++) {
              field = fields.get(i);                                    FieldName.append('`'+field.get
Fieldname().toLowerCase()+'`');
              FieldValue.append("'"+field.getValue()+"'");
              if (i ! = fields.size() - 1) {
                     FieldName.append(',');
                     FieldValue.append(',');
              }
        }
        insert_string.append("insert "+ record.getTablename()+"("+FieldName.toString()+") values("+Fi
eldValue.toString()+");");
}
```

Parse an UPDATE statement

If an UPDATE operation is performed in the source database, the operation type of the data record is UPDATE. The field values prior to the UPDATE operation are stored in `Record.getFieldList()` entries with even indexes. The field values after the UPDATE operation are stored in Record.getFieldList() entries with odd indexes.

If the UPDATE operation is performed on a table that has a primary key, you can use the following sample code to parse the UPDATE statement:

```
StringBuilder update_string=new StringBuilder();
Record.Type type=record.getOpt();
DataMessage.Record.Field field;
StringBuilder SetValue = new StringBuilder();
StringBuilder WhereCondition = new StringBuilder();
String ConditionStr;
boolean hasPk=false;
boolean pkMode=false;
boolean hasSet=false;
if(type.equals(Record.Type.UPDATE)){
    int i=0;
    DataMessage.Record.Field OldField = null;
    DataMessage.Record.Field NewField = null;
    List<DataMessage.Record.Field> fields = record.getFieldList();
    for (; i <fields.size() ; i++) {
        if (i % 2 == 0) {
            OldField = fields.get(i);
            continue;
        }
    NewField = fields.get(i);
    field = NewField;
    if (field.isPrimary()) {
        if (hasPk) {
            WhereCondition.append(" and ");
        }
        //where old value
```

```
        ConditionStr = getFieldValue(OldField);
        if(ConditionStr==null){                                    WhereCondition.append("`"+fie
ld.getFieldname().toLowerCase()+"`" + " " + "is null");
        }else{
               WhereCondition.append("`"+field.getFieldname().toLowerCase()+"`"+" = "+ "'"+OldField.g
etValue()+"'");
         }
        hasPk = true;
    }
    if (hasSet) {
        SetValue.append(",");
    }
      SetValue.append("`"+field.getFieldname().toLowerCase()+"`" + " = " + "'"+field.getValue()+"'");
    String setStr = getFieldValue(field);
    hasSet = true;
    }
    update_string.append("Update "+record.getTablename() +" Set " + SetValue + " Where "+WhereConditi
on +";");
}
protected String getFieldValue(Field field) throws Exception {
    ByteString byteString = field.getValue();
    if (byteString == null) {
        return null;
    }
    else {
        String value;
        if (field.getType() == com.aliyun.drc.client.message.DataMessage.Record.Field.Type.STRING &&
field.getEncoding() ! = null && field.getEncoding() ! = "ASCII") {
            value = field.getValue().toString(field.getEncoding());
        }
        else {
          value = byteString.toString();
        }
        return value;
    }
}
```

Parse a DELETE statement

If a DELETE operation is performed in the source database, the operation type of the data record is DELETE. If the DELETE operation is performed on a table that has a primary key, you can use the following sample code to parse the DELETE statement:

```
StringBuilder delete_string=new StringBuilder();
Record.Type type=record.getOpt();
DataMessage.Record.Field field;
StringBuilder FieldName=new StringBuilder();
StringBuilder FieldValue = new StringBuilder();
StringBuilder DeleteCondition = new StringBuilder();
boolean hasPk=false;
boolean pkMode=false;
if(type.equals(Record.Type.DELETE)){
   int i=0;
   List<DataMessage.Record.Field> fields = record.getFieldList();
   delete_string.append("Delete From" + record.getTablename() + "where");
   // Check whether the table has a primary key.
   if (record.getPrimaryKeys() ! = null) {
             pkMode = record.getPrimaryKeys().length() > 0 ? true : false;
   }
   for (; i < fields.size(); i++) {
            if ((pkMode && ! field.isPrimary())) {
                    continue;
            }
            if (hasPk) {
                    delete_string.append(" and ");
            }
            delete_string.append(field.getFieldname() + "=" + field.getValue());
            hasPk = true;
    }
    delete_string.append(";");
}
```

Parse a REPLACE statement

If a REPLACE operation is performed in the source database, the operation type of the data record is UPDATE or INSERT.

- If the value specified in the REPLACE statement does not exist, the operation type of the data record is INSERT.
- If the value specified in the REPLACE statement exists, the operation type of the data record is UPDATE.

Parse a BEGIN statement

If a BEGIN operation is performed in the source database, the operation type of the data record is BEGIN. You do not need to perform operations on fields because the BEGIN statement does not modify fields. You only need to check that the operation is a BEGIN operation. You can use the following sample code to parse the BEGIN statement:

```
StringBuilder sql_string = new StringBuilder();
Record.Type type = record.getOpt();
if(type.equals(Record.Type.BEGIN)){
        sql_string.append("Begin");
}
```

Parse a COMMIT statement

If a COMMIT operation is performed in the source database, the operation type for the data record is COMMIT. You do not need to perform operations on fields because the COMMIT statement does not modify fields. You only need to check that the operation is a COMMIT operation. You can use the following sample code to parse the COMMIT statement:

```
StringBuilder sql_string = new StringBuilder();
Record.Type type = record.getOpt();
if(type.equals(Record.Type.COMMIT)){
        sql_string.append("commit");
}
```

18.1.5.5.4. Run the SDK demo code

This section describes how to run the demo code provided by the DTS console.

1. Create an AccessKey.

   Your account must pass the AccessKey authentication before you can use an SDK to connect to a subscription channel. Therefore, before using the SDK, you must obtain an AccessKey. For more information, see the "**Obtain an AccessKey**" section of the *DTS Developer Guide* .

2. Install the Java SDK.

   The development environment supported by the DTS Java SDK is J2SE Development Kit (JDK) V1.5 or later.

   For an Eclipse project, you can follow these steps to install the Java SDK:

   i. Click **View Example Code** and download the SDK package *consumer.jar*.

   ii. Import the JAR package to an Eclipse project as follows:

   In Eclipse, right-click your project and choose **Properties > Java Build Path > Libraries > Add External JARs**. Select the path for storing the *consumer.jar* package consumer.jar.

   iii. Select the *consumer.jar* package and click **OK**.

   Then you can use the DTS Java SDK in the project.

3. Run the demo code.

   DTS provides the SDK demo code. You can copy the demo code by using the View Demo Code option in the DTS console. For an Eclipse project, you can follow these steps to run the demo code:

   i. Create a class named MainClass in the src directory of the Eclipse project.

   ii. Open the generated Java file *MainClass* and delete the code template.

   iii. Paste the demo code into the *MainClass* file.

   iv. Modify the AccessKeyId, AccessKeySecret, and subscription channel ID in the demo code.

   Change the marked parts in the preceding demo code to the AccessKeyId, AccessKeySecret, and subscription channel ID of your account.

   You can obtain the subscription channel ID from the DTS console.

   v. In Eclipse, right-click the demo file and choose **Run as > Java Application** to run the demo code.

18.1.5.6. Use a Kafka client to consume tracked data

This topic describes how to use the demo code of a Kafka client to consume tracked data. The change tracking feature of the new version allows you to consume tracked data by using a Kafka client from V0.11 to V1.1.

Prerequisites

- A change tracking task is created. For more information, see Track data changes from a user-created MySQL database or an ApsaraDB RDS for MySQL instance (new version) or Track data changes from a PolarDB-X instance.
- One or more consumer groups are created. For more information, see Create a consumer group.

Download and run the demo code of the Kafka client

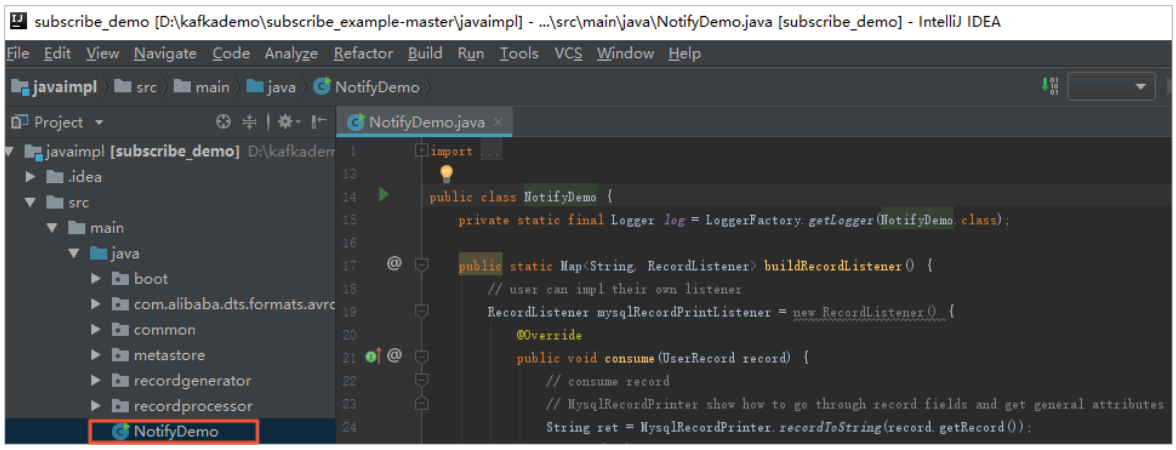
Click here to download the demo code of the Kafka client.

Download and run the demo code of the Kafka client

Step	File or directory
1. Use the native Kafka consumer to obtain incremental data from the change tracking instance.	subscribe_example-master/javaimpl/src/main/java/recordgenerator/
2. Deserialize the image of the incremental data, and obtain attributes such as the pre-image and post-image.	subscribe_example-master/javaimpl/src/main/java/boot/MysqlRecordPrinter.java
3. Convert the dataTypeNumber values in the deserialized data into MySQL or Oracle data types. **Note** For more information, see Mappings between MySQL data types and dataTypeNumber values and Mappings between Oracle data types and dataTypeNumber values.	subscribe_example-master/javaimpl/src/main/java/recordprocessor/mysql/

Procedure

1. Download the demo code of the Kafka client, and then decompress the package.
2. Open IntelliJ IDEA. In the window that appears, click **Open**.
3. In the dialog box that appears, go to the directory in which the downloaded demo code resides. Find the *pom.xml* file.
4. In the dialog box that appears, select **Open as Project**.
5. On the IntelliJ IDEA page, expand folders to find the demo file of the Kafka client, and double-click the file. The file name is *NotifyDemo.java*.



6. Set the parameters in the *NotifyDemo.java* file.

Parameter	Description	Method to obtain

Parameter	Description	Method to obtain
USER_NAME	The username of the consumer group. **Warning** If you are not using the Kafka client that is described in this topic, you must specify the username in the following format: `<Consumer group account>-<Consumer group ID>`, for example, `dtstest-dtsae******bpv`. Otherwise, the connection fails.	In the DTS console, click the instance ID, and then click **Data Consume**. You can obtain the **Consumer Group ID** and the corresponding **Account** information. **Note** The password of the consumer group account is specified when you create a consumer group.
PASSWORD_NAME	The password of the account.	
SID_NAME	The ID of the consumer group.	
GROUP_NAME	The name of the consumer group. Set this parameter to the consumer group ID.	
KAFKA_TOPIC	The topic of the change tracking task.	In the DTS console, click the instance ID. On the **Track Data Changes** page, you can obtain the **tracked topic**, network address, and port number.
KAFKA_BROKER_URL_NAME	The network address and port number of the change tracking task. **Note** If you track data changes over internal networks, the network latency is minimal. This is applicable if the ECS instance where you deploy the Kafka client belongs to the same VPC or classic network as the change tracking instance.	
INITIAL_CHECKPOINT_NAME	The consumer offset of consumed data. The value is a UNIX timestamp. **Note** You must save the consumer offset. If the consumption process is interrupted, you can specify the consumer offset on the change tracking client to resume data consumption. This allows you to prevent against data loss. When you start the change tracking client, you can specify the consumer offset to consume data on demand.	When you use the Kafka client to track data changes for the first time, convert the required time point into a UNIX timestamp.

Parameter	Description	Method to obtain
USE_CONFIG_CHECKPOINT_NAME	Default value: *true*. The default value indicates that the client is forced to consume data from the specified consumer offset. This allows you to retain the data that is received but not processed.	None.

7. On the top of the IntelliJ IDEA page, choose **Run > Run** to run the client.

> **Note** When you run IntelliJ IDEA for the first time, it loads and installs the relevant dependency.

Mappings between MySQL data types and dataTypeNumber values

Mappings between Oracle data types and dataTypeNumber values

19.Data Management (DMS)

19.1. User Guide

19.1.1. What is DMS?

Data Management (DMS) is a fully managed service that is provided by Alibaba Cloud. This service allows you to manage data, schema, development procedures, development specifications, users, and permissions. DMS also provides security control to ensure secure access to databases.

Supported databases

- Relational databases: MySQL, SQL Server, PostgreSQL, PolarDB-X (previously called DRDS), Oracle, and ApsaraDB for OceanBase.
- NoSQL databases: KVStore for Redis and ApsaraDB for MongoDB.
- Analytical databases: AnalyticDB for MySQL and AnalyticDB for PostgreSQL.

Features

- DMS provides support for the entire process of database development. The process includes the following stages: 1. Design table structures in an on-premises environment based on the predefined design specification. 2. Publish and produce SQL reviews that are included in code and schemas to a specified environment on demand. The preceding operations are performed before the code is released. These SQL reviews in code are used to add, remove, modify, or query rows.
- DMS provides fine-grained access control at the database, table, and field levels. You can perform all the required operations on databases in the DMS console. These operations can be traced and audited.
- DMS allows you to configure the required operation specifications and approval processes for multiple modules. These modules include the schema design, data changes, data export, and permission requests.
- DMS provides an integrated platform that connects database development with database interaction. You can manage databases without the need to use database accounts and passwords to switch between database endpoints at a high frequency.
- DMS provides the task orchestration feature. This feature allows you to orchestrate and schedule SQL tasks for databases on a regular basis. You can use this feature to perform the required operations with ease. For example, you can transfer historical data, analyze periodical reports, and generate analytical results.

19.1.2. Quick start

19.1.2.1. Log on to the DMS console

This topic describes how to log on to the DMS console by using Google Chrome.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

> **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
>
> - Uppercase or lowercase letters
> - Digits
> - Special characters, which include ! @ # $ %

3. Click **Login**.
4. In the top navigation bar, move the pointer over **Products** and click **Data Management** under **Database Services**.
5. Set the **Organization** and **Region** parameters and click **DMS**.

> **Note** If you log on to the DMS console as a DMS administrator and your account is added to multiple tenants, you can move the pointer over the
>
> 
>
> icon in the upper-right corner and click **Switch tenant** to switch to another tenant.

19.1.2.2. Register database instances with DMS

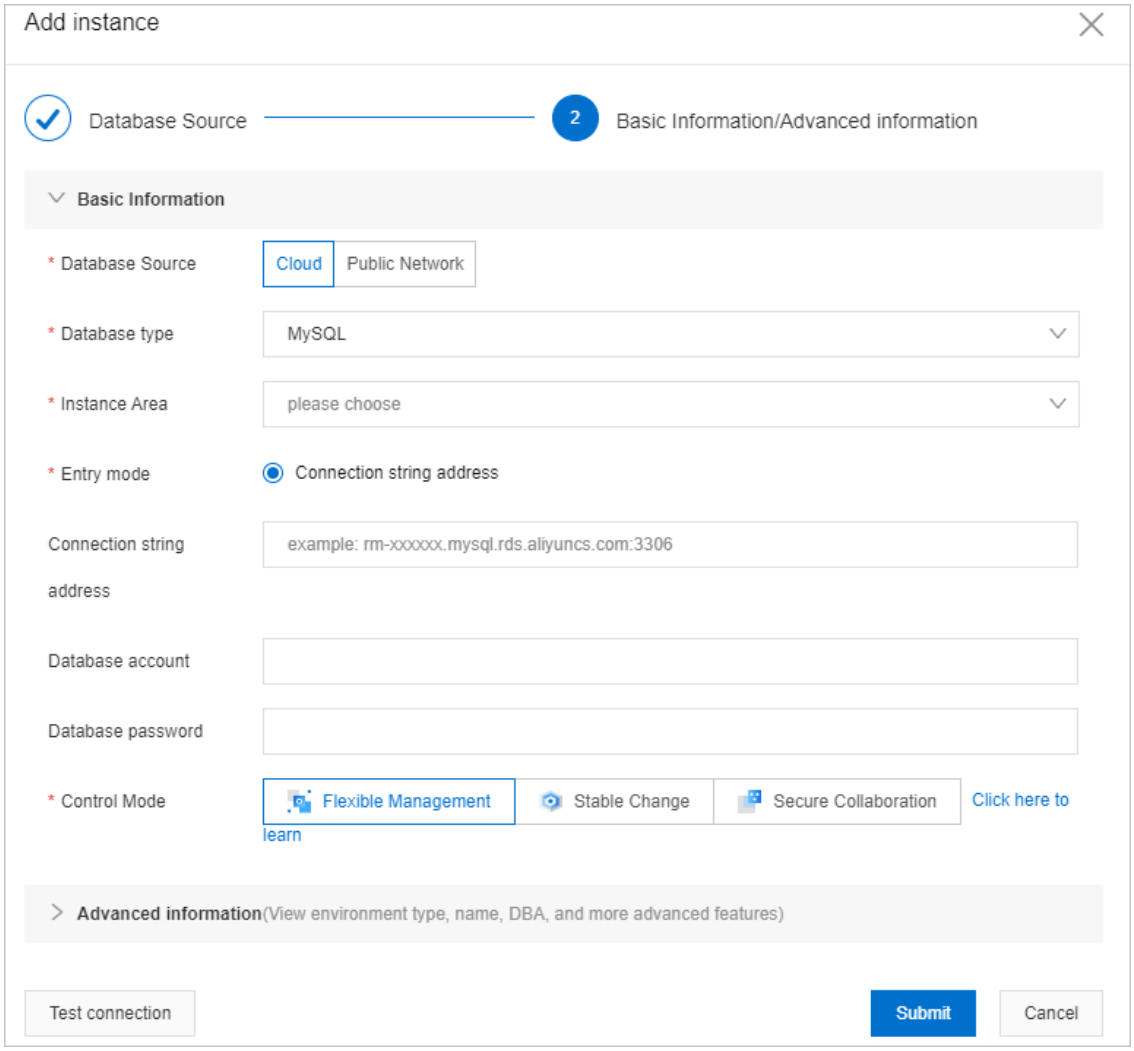
To manage database instances in DMS, you must register the database instances with DMS. DMS allows you to register ApsaraDB instances and self-managed database instances that are hosted over the Internet. This topic shows you how to register an ApsaraDB RDS for MySQL instance with DMS.

Procedure

1. Log on to the DMS console.
2. In the left-side navigation pane, move the pointer over **Add instance / Batch entry** and select Add instance.

> **Note** You can also move the pointer over the + icon in the left-side navigation pane and select Add instance.

3. In the Add instance dialog box, click the **Cloud** tab.
4. On the Cloud tab, select a database type.
5. In the Add instance dialog box, set the parameters as required. This example shows you how to register an ApsaraDB RDS for MySQL instance with DMS.



Section	Parameter	Description
Basic Information	**Database source**	The source of the database instance. In this example, select **Cloud**.
	Database type	The type of the database instance. In this example, select **MySQL**.
	Instance Area	The region where the database instance resides.
	Entry mode	The method that you use to log on to the database instance. Default value: **Connection string address**. This value cannot be changed.
	Connection string address	The endpoint of the database instance. The endpoint contains information about a port number.
	Database account	The username that you use to log on to the database instance.
	Database password	The password that you use to log on to the database instance.

Section	Parameter	Description
	Control Mode	The control mode that is used to manage the database instance. For more information, see Control modes. **Note** If you set this parameter to **Security Collaboration**, you must set the **Security Rules** parameter.
Advanced information	**Environment type**	The environment of the database instance.
	Instance Name	The name that you specify for the instance.
	Enable DSQL	Specifies whether to enable the cross-database query feature. To enable the cross-database query feature, you must specify a database link name. For more information, see Cross-database query.
	OnlineDDL	Specifies whether to allow the database instance to change schemas without locking tables.
	DBA	The database administrator (DBA) of the database instance. The DBA can grant permissions to users.
	query timeout(s)	The timeout period for the execution of an SQL query statement. If the execution of an SQL query statement lasts longer than the specified timeout period, the execution of the statement is terminated to protect the database.
	export timeout(s)	The timeout period for the execution of an SQL export statement. If the execution of an SQL export statement lasts longer than the specified timeout period, the execution of the statement is terminated to protect the database.

6. After you complete the configurations, click **Test connection** in the lower-left corner of the **Basic Information** section.

   Note If the connectivity test fails, check the specified parameter values based on the error message.

7. Click **Submit**.

Result

After the preceding steps are performed, the ApsaraDB instance is registered with DMS. You can view and manage your database instance in the left-side navigation pane of the DMS console.

19.1.2.3. Add a user

DMS provides the user management feature that allows you to add users and assign roles based on your business requirements.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **System Management > User**.

> **Note** On the User tab, you can edit, enable, disable, and remove existing users.

3. Click **New**.
4. In the Add User dialog box, set the parameters that are described in the following table.

Parameter	Description
Alibaba Cloud Account	The ID of an Apsara Stack tenant account or a RAM user. You can enter one of the following IDs: ○ The ID of another Apsara Stack tenant account. You can obtain this ID from the account owner. ○ The ID of a RAM user. You can view this ID in the Apsara Uni-manager Operations Console.
Role	The role that you want to assign to the user based on your business requirements. You can assign one or more of the following roles: ○ A regular user ○ A DBA ○ A DMS administrator ○ A security administrator **Note** For more information about the features that are supported by each role, see Features that are supported by each role.

5. Click **OK**.

19.1.3. Control modes

DMS provides three control modes for you to manage instances: Flexible Management, Stable Change, and Security Collaboration. You can specify a control mode for each instance.

Control mode	Description	Scenario	Logon method
Flexible management	This control mode allows you to manage the visualized data and schemas of multiple types of databases. It also provides a variety of data management solutions. This simplifies the use of databases and facilitates management.	• Database instances do not require strict control. • Database instances are used by a single user.	A database account and the related password.
Stable change	• This control mode provides multiple solutions to ensure database reliability. These solutions allow you to change data without the need to lock the related table or schema. • All features that are included in the flexible management control mode are available.	• Database instances require a high level of availability. This ensures that these database instances function as expected for an extended period of time. • Database instances are used by a small-sized group that includes multiple users.	A database account and the related password.

Control mode	Description	Scenario	Logon method
Security collaboration	• This control mode provides multiple solutions to ensure data security. These solutions include fine-grained access control at the database, table, or field level and sensitive data management. • This control mode allows you to produce enterprise-specific database DevOps solutions through custom design specifications and approval processes. • All features that are included in the flexible management and stable change control modes are available.	• Ensure the data security of database instances. • Implement strict access control over development or change workflows. • Manage compliance for enterprises.	Logon-free through authorization.

Note The instances that are managed in Stable Change mode consume the billing quota of the instances that are managed in Security Collaboration mode.

19.1.4. Features that are supported by each role

DMS provides the following roles: regular user, DBA, security administrator, and DMS administrator. This topic describes the features that are supported by each role.

Category	Feature	Regular user	DBA	Security administrator	DMS administrator	Description
Permission	Permission management	√	√	√	√	You can use this feature to apply for permissions on instances, databases, tables, and sensitive fields. You can also view permissions that you have.
	Data Changes	√	√	√	√	You can use this feature to initialize data for a newly published project, clean up historical data, fix bugs, or run a test.
	Data Import	√	√	√	√	You can use this feature to import a large amount of data to your databases at a time.
	Data Export	√	√	√	√	You can use this feature to export a large amount of data for analysis or export the required data.
	Data Tracking	√	√	√	√	If specific data fails to meet your requirements due to reasons such as misoperation, you can use this feature to restore data to the normal state.

Category	Feature	Regular user	DBA	Security administrator	DMS administrator	Description
Data Plans	Test Data Generate	√	√	√	√	Some business scenarios may require frequent data preparation. In this case, you can use this feature to generate test data to ensure data security and discreteness and improve production efficiency.
	Data Warehouse Development	√	√	√	√	DMS uses a database as a computing engine and integrates various tools and services, such as Data Transmission Service (DTS) and Data Lake Analytics (DLA), in the database ecosystem for data warehouse development. You can use this feature to develop and manage data warehouses in DMS with ease.
	Data Service	√	√	√	√	You can use this feature to export data at the field or row level, display data in a visualized manner, and publish API operations to the Alibaba Cloud Marketplace for sale.
	Database Clone	√	√	√	√	You can use this feature to clone MySQL databases.
Schemas	Schema Design	√	√	√	√	When you develop or optimize projects or process new business requirements, you can use this feature to change schemas. For example, you can use this feature to create a table or modify an existing table.
	Table Sync	√	√	√	√	You can use this feature to compare and synchronize the schemas of tables in different environments, such as online and offline environments. This feature helps ensure the consistency of schemas.
Optimization	SQL Review	√	√	√	√	You can use this feature to prevent SQL statements that do not use indexes or do not conform to database development standards. This feature helps protect against SQL injection attacks.
SQLConsole	Single Database query	√	√	√	√	You can write SQL statements to query data in a single database. This feature can be used to verify business code, analyze product effects, and identify issues in an online environment.

Category	Feature	Regular user	DBA	Security administrator	DMS administrator	Description
	Cross-database Query	√	√	√	√	You can use this feature to perform join queries across online heterogeneous databases that are deployed in different environments.
System Management	Instance management	×	√	×	√	You can use this feature to manage instances. For example, you can register, view, or edit instances.
	User management	×	×	×	√	You can use this feature to manage users. For example, you can add, view, or edit users as needed.
	Task management	×	√	×	√	You can use this feature to manage tasks. For example, you can create, start, or stop tasks.
	Configuration management	×	×	×	√	You can use this feature to view and modify system configurations, or view the historical modifications of the configurations.
Security management	Security Rules	×	√	×	√	You can use this feature to configure security rules. Only SQL statements that conform to the security rules can be executed.
	Approval Processes	×	√	×	√	Approval processes are associated with security rules. You can configure different approval processes for different types of tickets.
	Operation Logs	×	√	√	√	Operations logs record data changes. Each record contains information such as the user who performed the operation, operation details, and time at which the operation was performed. You can use this feature to track historical user operations at any time.
	Access IP Whitelists	×	×	×	√	After you configure an access IP whitelist, only the IP addresses or Classless Inter-Domain Routing (CIDR) blocks in the whitelist can access the resources within your DMS tenant. This effectively enhances data security.
	Sensitive Data	×	√	√	√	You can use this feature to manage sensitive data. For example, you can use algorithms to de-identify sensitive data or adjust the security levels of sensitive data.

Category	Feature	Regular user	DBA	Security administrator	DMS administrator	Description
Tickets	Ticket management	√	√	√	√	You can use this feature to configure notification methods. DMS can notify you of the approval or execution status of tickets by using DingTalk notifications or emails.

19.1.5. Apply for permissions

In DMS, you can apply for the query, change, or export permission on a database, table, or column. After the database owner approves your application, you can perform data query, change, or export operations.

Permissions

- Query permission: the permission to execute SQL statements in the SQLConsole to query the data of the object on which you want to apply for the permission.
- Change permission: the permission to submit tickets to change data or synchronize data in a database or table. You cannot change data without approval.
- Export permission: the permission to submit tickets to export data from the object on which you want to apply for the permission. You cannot export data without approval.

Permission categories that are supported by each control mode

Permission category	Description	Control mode		
		Flexible Management	Stable Change	Security Collaboration
Instance-Login	You must first obtain the permission to access an instance and then use the preset database account and password to log on to the instance.	√	√	×
Database-Permission	Database permissions are divided into three types: query, export, and change permissions. After you are granted permissions on a database, you have access to all data in the database except sensitive fields and the tables where specific data rows are managed. You also have access to tables that are newly created in the database. • Query permission: You can execute SQL statements in the SQLConsole to query data. • Change permission: You can submit data change and data import tickets. • Export permission: You can submit data export tickets.	×	×	√

Permission category	Description	Control mode		
		Flexible Management	Stable Change	Security Collaboration
Table-Permission	Table permissions are divided into three types: query, export, and change permissions. After you are granted permissions on a table, you have access to all data in the table except sensitive fields. • Query permission: You can execute SQL statements in the SQLConsole to query data. • Change permission: You can submit data change and data import tickets. • Export permission: You can submit data export tickets.	×	×	√
Sensitive Column-Permission	Sensitive field permissions are divided into three types: query, export, and change permissions. After you are granted permissions on sensitive fields in a table, you have access to all data in the table including sensitive fields. Before you apply for permissions on specific sensitive fields, you must have access to the database and table that contain the sensitive fields. • Query permission: You can execute SQL statements in the SQLConsole to query data. • Change permission: You can submit data change and data import tickets. • Export permission: You can submit data export tickets.	×	×	√
Database-OWNER	• The owner of a database can manage permissions on the database. For example, the owner of a database can grant or revoke permissions on the database or the tables in the database. • The owner of a database can query all data in the database except sensitive or confidential fields. The owner of a database can also submit tickets to perform operations on the data and schemas in the database without the need to apply for permissions. • DMS automatically identifies database owners and then assign them to the owner nodes in approval processes.	√	√	√
Table-OWNER	• The owner of a table can manage permissions on the table. For example, the owner of a table can grant or revoke permissions on the table. • The owner of a table can query all data in the table except sensitive or confidential fields.	√	√	√

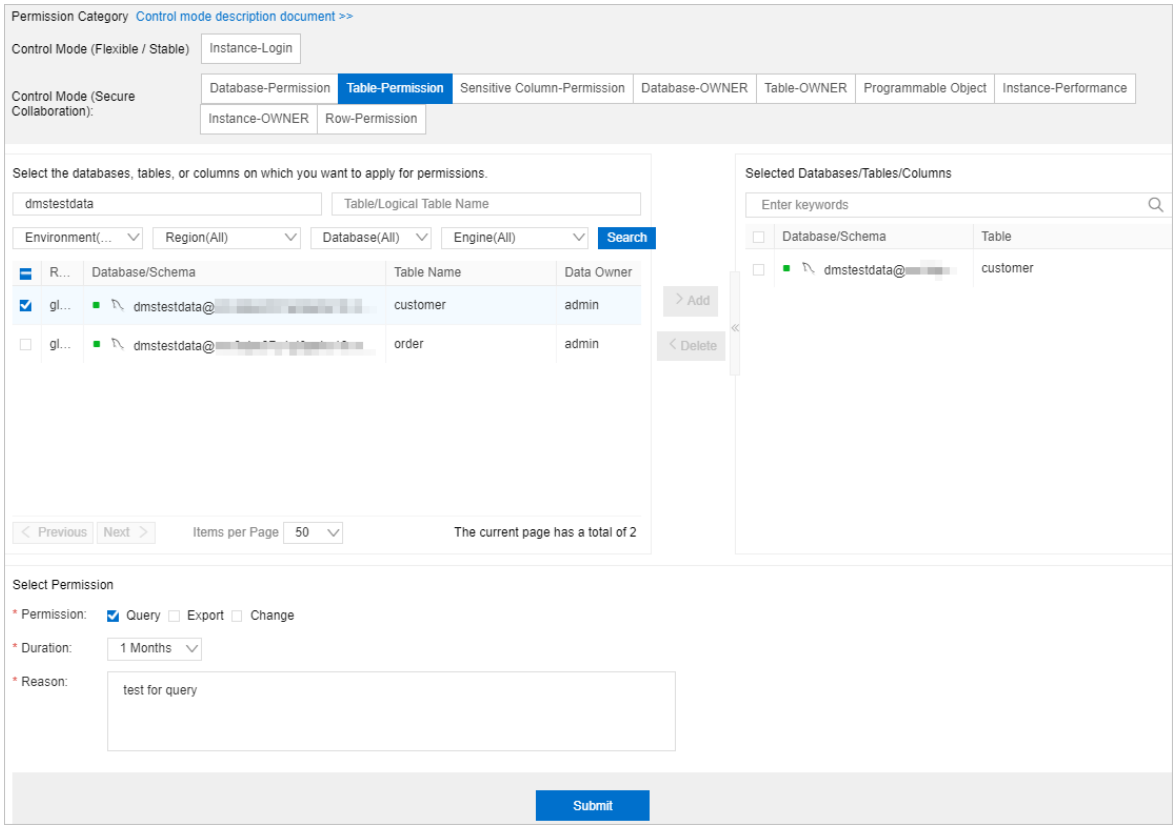
Permission category	Description	Control mode		
		Flexible Management	Stable Change	Security Collaboration
Programmable Object	Permissions on programmable objects are divided into three types: query, export, and change permissions. • Query permission: You can execute SQL statements in the SQLConsole to query data. • Change permission: You can submit data change and data import tickets. • Export permission: You can submit data export tickets.	×	×	×
Instance-Performance	You can apply for permissions to view the performance of instances that are managed in Security Collaboration mode.	×	×	√
Instance-OWNER	• The owner of an instance can manage permissions on the instance. For example, the owner of an instance can grant or revoke permissions on the instance. • The owner of an instance can query all data in the databases of the instance except sensitive or confidential fields. The owner of an instance can also submit tickets to perform operations on the data and schemas in the instance without the need to apply for permissions.	√	√	√
Row-Permission	Row permissions are divided into three types: query, export, and change permissions. You can apply for permissions on specific values of a managed field in a table. You can also apply for permissions on all values of a managed field in a table. • Query permission: You can execute SQL statements in the SQLConsole to query data. • Change permission: You can submit data change and data import tickets. • Export permission: You can submit data export tickets.	×	×	×

Apply for permissions

1. Log on to the DMS console.
2. In the top navigation bar, choose **Permission > Apply Permission** and select the permission category for which you want to apply. For more information about permission categories, see the Permission categories that are supported by each control mode table in this topic.

   > **Note** You can also enter a database name or a table name in the search box of the top navigation bar to search for a database or table. In the search results, find the database or table on which you want to apply for permissions and click **Access apply** in the **Actions** column.

3. Configure the information about the permission for which you want to apply.



i. Select the category of the permission for which you want to apply.

ii. Select the databases, tables, or columns on which you want to apply for permissions.

> **Note** Enter keywords, specify filter conditions, and then click Search to search for databases or tables. The keywords that you enter can contain percent signs (%) as wildcards. In the search results, select the databases or tables on which you want to apply for permissions and click Add.

iii. Select the type of permission for which you want to apply, specify the duration for which you want to have the permission, and then enter the reason for applying for the permission.

4. Click **Submit** and wait for approval.

> **Note** You can view the status of the ticket in the My Tickets section of the **Workbench** tab.

Manage permissions

Management type	Operation	Description
Active management	Release permissions	On the Workbench tab, click **Effective Permissions** in the Permissions section. Select the object for which you want to release permissions and click **Release Permission**.
	Renew permissions	On the Workbench tab, click **Expiring Permissions** in the Permissions section to view and check the permissions that will expire soon. If you want to renew a specific permission, submit a ticket to apply for the permission.
Passive management	N/A	The owner of a database can view and check the rationality of permissions that are granted to users at any time and manage the permissions.

> **Note** Assume that you have applied for, released, revoked, or granted permissions. You can view all these permission operations in operations logs. To view the operations logs, choose **System Management** > **Security** > **Operation Logs** in the top navigation bar.

19.1.6. Data plans

19.1.6.1. Change data

DMS provides the data change module that allows you to change data. This topic describes how to use this module to change data.

Context

DMS allows you to submit data change tickets to initialize data for a newly published project, clear historical data, fix bugs, or run a test. The operations that you can perform to change data include, but are not limited to, insert, update, delete, or truncate operations.

Data change features

Feature	Description
Normal Data Modify	You can use the Normal Data Modify feature to perform the following data changes: • Perform normal data changes. • Perform lock-free schema changes. You can perform such operations to change character sets and collations for tables, adjust time zones, and change column data types. Compared with normal data change operations, such operations can be performed to achieve the following benefits: ◦ Allows you to change schemas without affecting business. ◦ Avoids latency in synchronization between primary and secondary databases that occurs when schemas are changed by using native online data definition language (DDL) operations. ◦ Reclaims tablespaces and reduces fragmentation rates without locking tables. You no longer need to use the OPTIMIZE TABLE statement that causes tables to be locked. **Note** You can use this feature only for MySQL databases. Before you use this feature, you must set the OnlineDDL parameter in the **Advanced information** section to Open(DMS OnlineDDL first) when you register or edit an instance. For more information, see Register database instances with DMS.
Lock-Free Data Modify	You can use this feature to change a large amount of data. For example, you can use this feature to delete historical data and update all fields in a table. The SQL statements for data changes are divided and executed in different batches based on the primary key or unique key. This limits the consumption of database performance and space. **Note** You can use this feature only for MySQL databases.
History Data Clean	You can use this feature to regularly clean historical data to prevent the accumulation of historical data from affecting the stability of the production environment. **Note** You can use this feature only for MySQL databases.

Feature	Description
Programmable Object	Databases provide programmable objects such as stored functions and stored procedures. This feature allows you to use the programmable objects to standardize management processes and provide audit records.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Data Plans > Data Changes** and select the data change feature that you want to apply.

   > **Note** This example shows you how to use the **Normal Data Modify** feature.

3. Set the parameters on the Data Change Ticket Application tab.

Parameter	Description
Reason Category	The reason for the data change. This helps you find the ticket in subsequent operations.
Business Background	The purpose or objective of the data change. This reduces unnecessary communication.
Change Stakeholder	The stakeholders of the data change. All specified stakeholders can view the ticket details and assist in the approval process. Irrelevant users other than DMS administrators and DBAs are not allowed to view the ticket details.
Execution Method	The way in which you want the ticket to be submitted for execution based on your business requirements.
Database	The database on which you have the change permission. You cannot submit a data change ticket if you have only permissions to query data in the database or change data in tables.
Affected Rows	The estimated number of data rows to be affected by the data change. To obtain the actual number of affected rows, you can use the COUNT function in SQL statements on the SQLConsole tab.
SQL Statements for Change	The executable SQL statements for changing data. You can write the SQL statements in the field or upload an SQL script to provide the SQL statements. DMS checks whether the syntax of the SQL statements is valid when you submit the ticket. If the syntax is invalid, you cannot submit the ticket.
SQL Statements for Rollback	The executable SQL statements for rolling back the data change operation. You can write the SQL statements in the field or upload an SQL script to provide the SQL statements.
Attachments	The images or files that are uploaded to add more information about the data change.

4. After you complete the configurations, click **Submit**.
5. After your ticket passes the precheck, click **Submit for Approval**. In the message that appears, click **OK**.
6. After the ticket is approved, click **Execute Change**.
7. Set the Execute Immediately parameter and click **Confirm Execution**.

> **Note** By default, the Execute Immediately switch is turned on. You can turn off **Execute Immediately** and specify a point in time for DMS to automatically run the task.

8. Wait until the execution is complete.

19.1.6.2. Import data

DMS provides the data import feature that allows you to import large amounts of data to a database in a quick manner. This helps you save manpower and resources.

Supported databases

- MySQL
- PolarDB-X

Supported file formats

- TXT files. Each TXT file for importing data can be up to 5 GB in size.
- SQL scripts. Each SQL script for importing data can be up to 5 GB in size.

> **Note** By default, you can use only INSERT and REPLACE statements to import data to database instances that are managed in Security Collaboration mode. If you want to use other SQL statements to import data, modify the security rules for data import as a DBA or DMS administrator. To modify the security rules, click the **SQL Correct** tab on the **Security Rules** tab and set the Checkpoints parameter to **Batch Data import rules**.

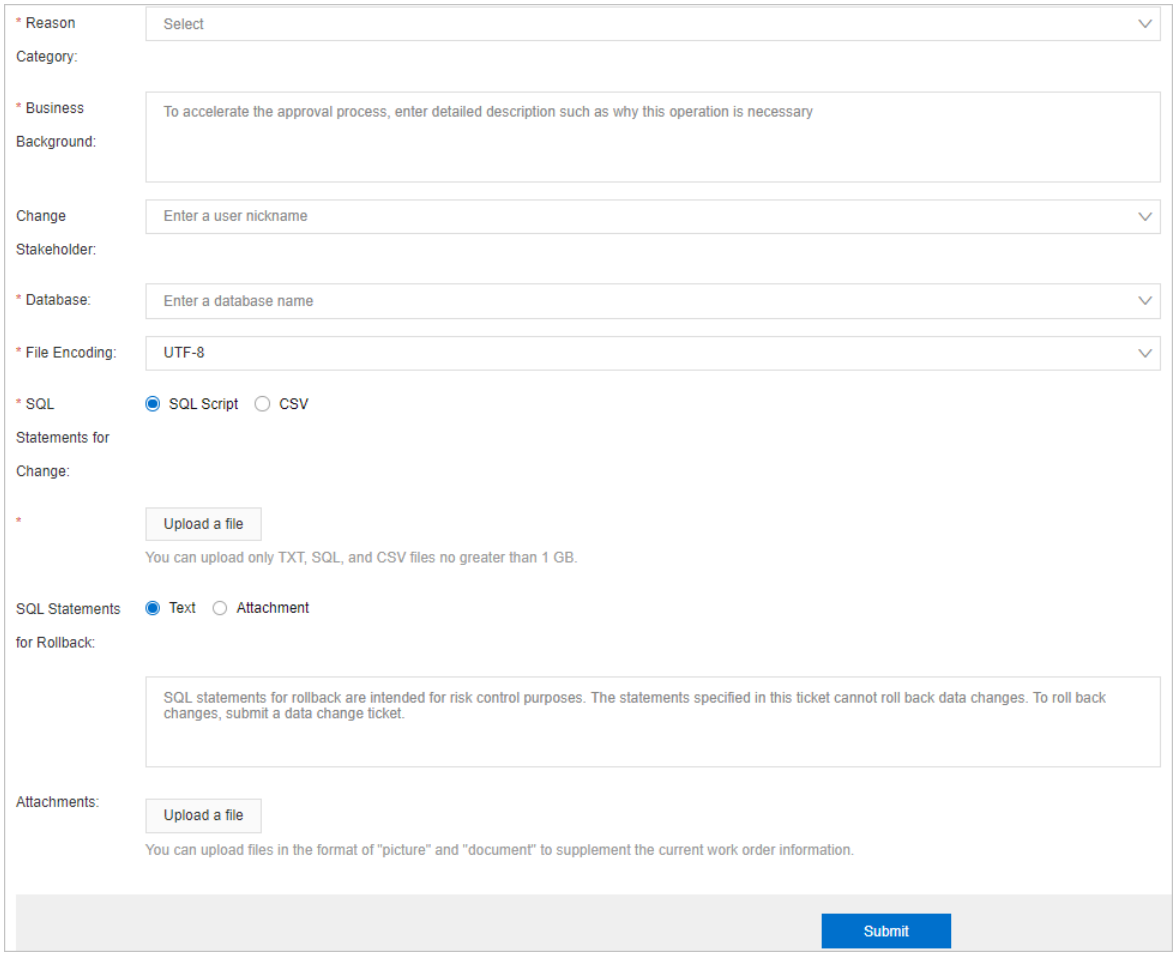
- CSV files. Values in a CSV file must be separated by commas (,). The first row must be field names.

Usage notes

- If you need to use SQL statements to import only a small amount of data, we recommend that you submit a Normal Data Modify or Lock-Free Data Modify ticket to ensure stable data change. For more information, see Change data.
- If you submit a Large Data Import ticket to import a large amount of data to a table, the table will be locked even if you set the OnlineDDL parameter to Open(DMS OnlineDDL first) for the database instance.
- We recommend that you use SQL statements with better performance to import a large amount of data, such as INSERT, UPDATE, and DELETE statements. Indexes of primary keys are used in the UPDATE and DELETE statements.
- By default, you can use only INSERT and REPLACE statements to import data to database instances that are managed in Security Collaboration mode. If you want to use other SQL statements to import data, modify the security rules for data import as a DBA or DMS administrator.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Data Plans > Data Import**.
3. On the Large Data Import tab, set the parameters that are described in the following table.



Parameter	Description
Reason Category	The reason for the data import. This helps you find the ticket in subsequent operations.
Business Background	The purpose or objective of the data import. This reduces unnecessary communication.
Change Stakeholder	The stakeholders of the data import. All specified stakeholders can view the ticket details and assist in the approval process. Irrelevant users other than DMS administrators and DBAs are not allowed to view the ticket details.
Database	The database on which you have the change permission. You cannot submit a data import ticket if you have only permissions to query data in the database or change data in tables.
File Encoding	The encoding algorithm to be used by the database.
SQL Statements for Change	The executable SQL statements for importing data. You can upload a file to provide the SQL statements. DMS checks whether the syntax of the SQL statements is valid when you submit the ticket. If the syntax is invalid, you cannot submit the ticket.
SQL Statements for Rollback	The executable SQL statements for rolling back the data import operation. You can write the SQL statements in the field or upload an SQL script to provide the SQL statements.
Attachments	The images or files that are uploaded to add more information about the data import.

Parameter	Description

4. After you configure the settings, click **Submit**.
5. After your ticket passes the precheck, click **Submit for Approval**. In the message that appears, click **OK**.
6. After the ticket is approved, click **Execute Change**.
7. Set the Execute Immediately parameter and click **Confirm Execution**.

> **Note** By default, the Execute Immediately switch is turned on. You can turn off the **Execute Immediately** switch and specify a point in time to run the ticket. The system automatically runs the ticket at the specified point in time.

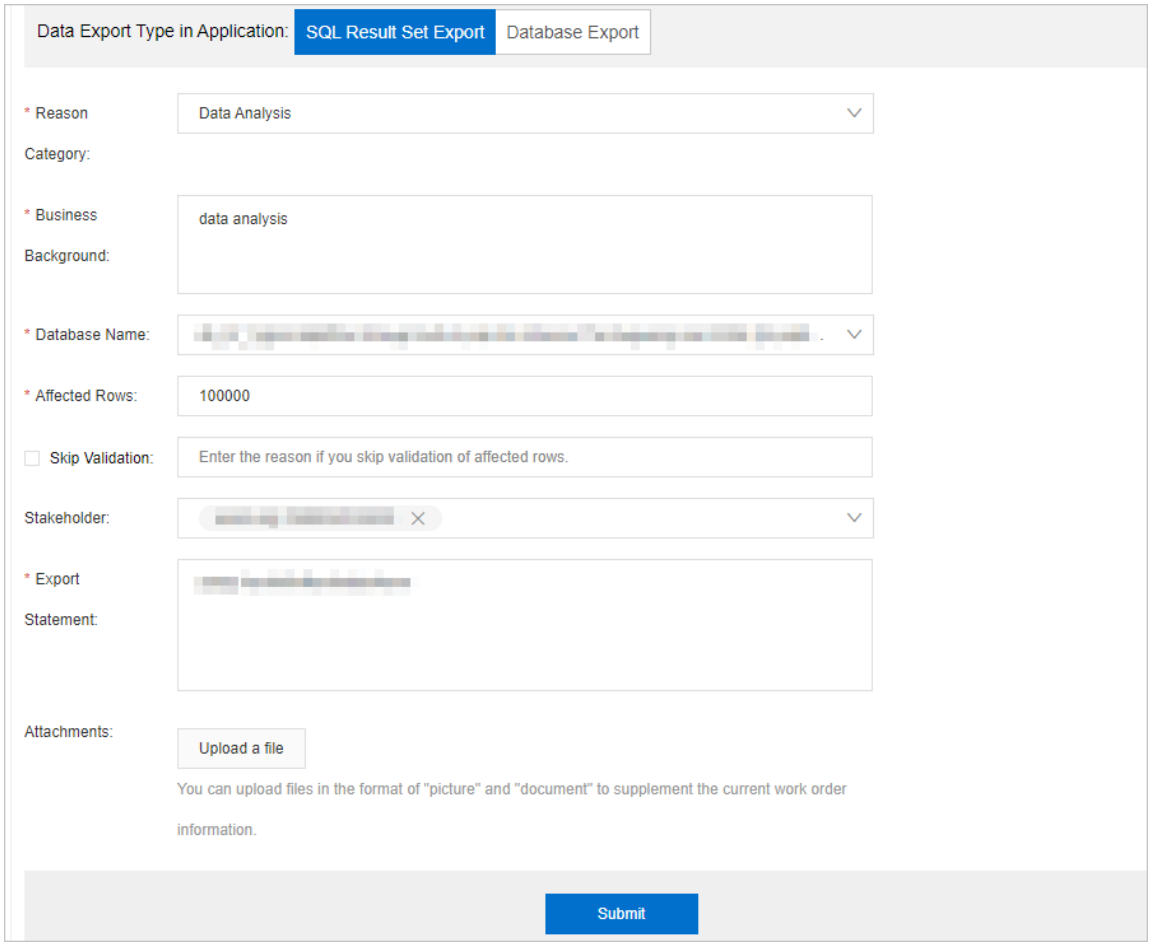
8. Wait until the execution is completed.

19.1.6.3. Export data

DMS provides the data export feature. You can use this feature to export a database or SQL result sets for data analysis.

Procedure

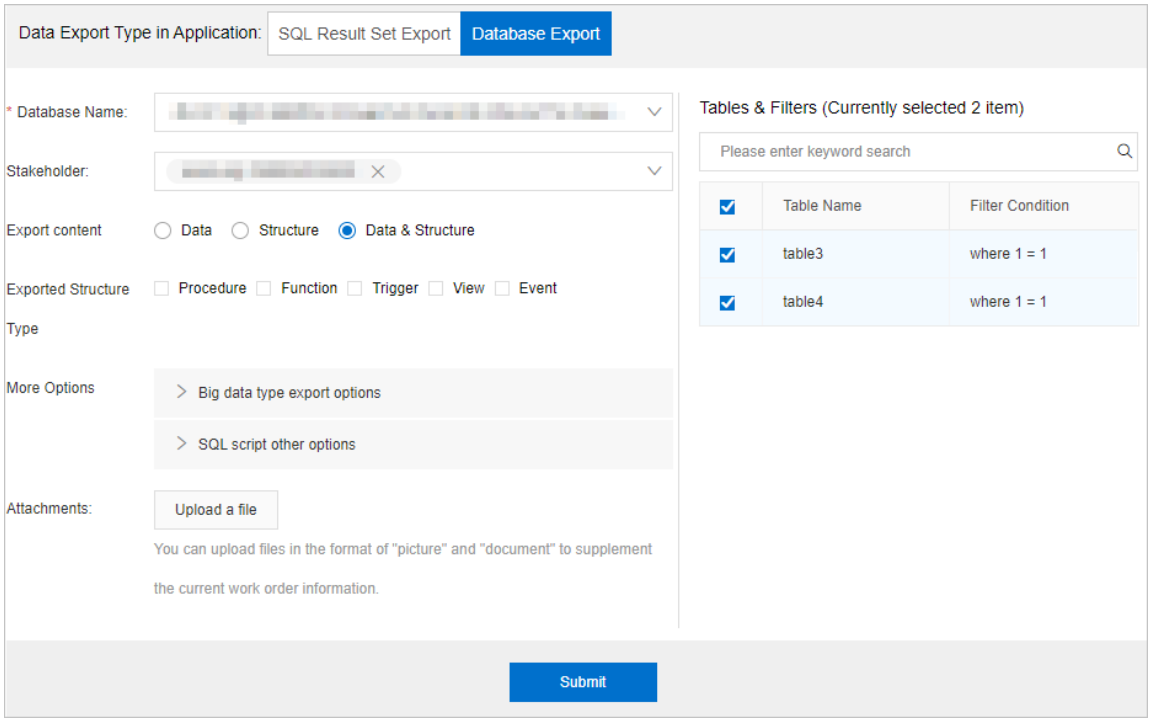
1. Log on to the DMS console.
2. In the top navigation bar, choose **Data Plans > Data Export**.
3. Select **SQL Result Set Export** or **Database Export** based on your business requirements.
4. On the Data Export Ticket Application tab, set the parameters as required.
   - Set the parameters on the SQL Result Set Export tab



Parameter	Description
Reason Category	The reason for the data export. This helps you find the ticket in subsequent operations.
Business Background	The purpose or objective of the data export. This reduces unnecessary communication.
Database Name	The database on which you have the export permission.
Affected Rows	The estimated number of data rows to be affected by the data export. To obtain the actual number of affected rows, you can use the `COUNT` function in SQL statements on the SQLConsole tab.
Skip Validation	Specifies whether to skip validation. If you select **Skip Validation**, you must enter a reason in the field next to the check box. **Warning** If you select **Skip Validation**, DMS does not check the number of rows to be affected by the data export. If a large amount of data is exported, your business may be affected. Proceed with caution.
Stakeholder	The stakeholders of the data export. All specified stakeholders can view the ticket details and assist in the approval process. Irrelevant users other than DMS administrators and DBAs are not allowed to view the ticket details.

Parameter	Description
Export Statement	The executable SQL statements for exporting data. Example: `select * from testtable`. DMS checks whether the syntax of the SQL statements is valid when you submit the ticket. If the syntax is invalid, you cannot submit the ticket.
Attachments	The images or files that are uploaded to add more information about the data export.

- Set the parameters on the Database Export tab



Parameter	Description
Database Name	The database on which you have the export permission. After you select the database, select the table where you want to export data and configure filter conditions in the Tables & Filters section.
Reason Category	The reason for the data export. This helps you find the ticket in subsequent operations.
Business Background	The purpose or objective of the data export. This reduces unnecessary communication.
Stakeholder	The stakeholders of the data export. All specified stakeholders can view the ticket details and assist in the approval process. Irrelevant users other than DMS administrators and DBAs are not allowed to view the ticket details.
Export content	The data content that you want to export based on your business requirements. Valid values: **Data**, **Structure**, and **Data & Structure**.
Exported Structure Type	The type of structure to be exported based on your business requirements.

Parameter	Description
More Options	The other objects that you want to export. Those objects are grouped into two export categories named **Big data type export options** and **SQL script other options**. You can click one of the categories and select a specific object as required.
Attachments	The images or files that are uploaded to add more information about the data export.

5. After you complete the configurations, click **Submit** and wait for approval.

   > **Note** When you export an SQL result set, DMS prechecks the SQL statements. After the SQL statements pass the precheck, click **Submit for Approval**. In the message that appears, click **OK**.

6. After your ticket is approved, go to the **Workbench** tab and click **Submitted Tickets** in the My Tickets section.
7. Find the data export ticket that you have submitted and click the ticket number.
8. In the **Execute/Automatic Execution** section, click **Download Exported File**.

19.1.6.4. Generate test data

DMS provides the Test Data Generate feature that allows you to generate data in a quick manner. You can generate test data for functional or performance tests.

Prerequisites

- A relational database is created to generate test data. The relational database may be a self-managed MySQL, ApsaraDB RDS for MySQL, AnalyticDB for MySQL, or PolarDB-X database.
- A table is created. You can use the Schema Design feature to create a table. For more information, see Design a schema.

Context

Functional tests or performance tests often require test data. In general, you may use the following methods to generate test data:

- Write test data. This method has low efficiency and is inapplicable to scenarios where a large amount of test data is required.
- Maintain existing scripts. This method incurs high costs. In addition, the data that is generated by using this method is not discrete enough.
- Use the data that is exported from an online environment as test data. This method is not secure and may cause data leak.

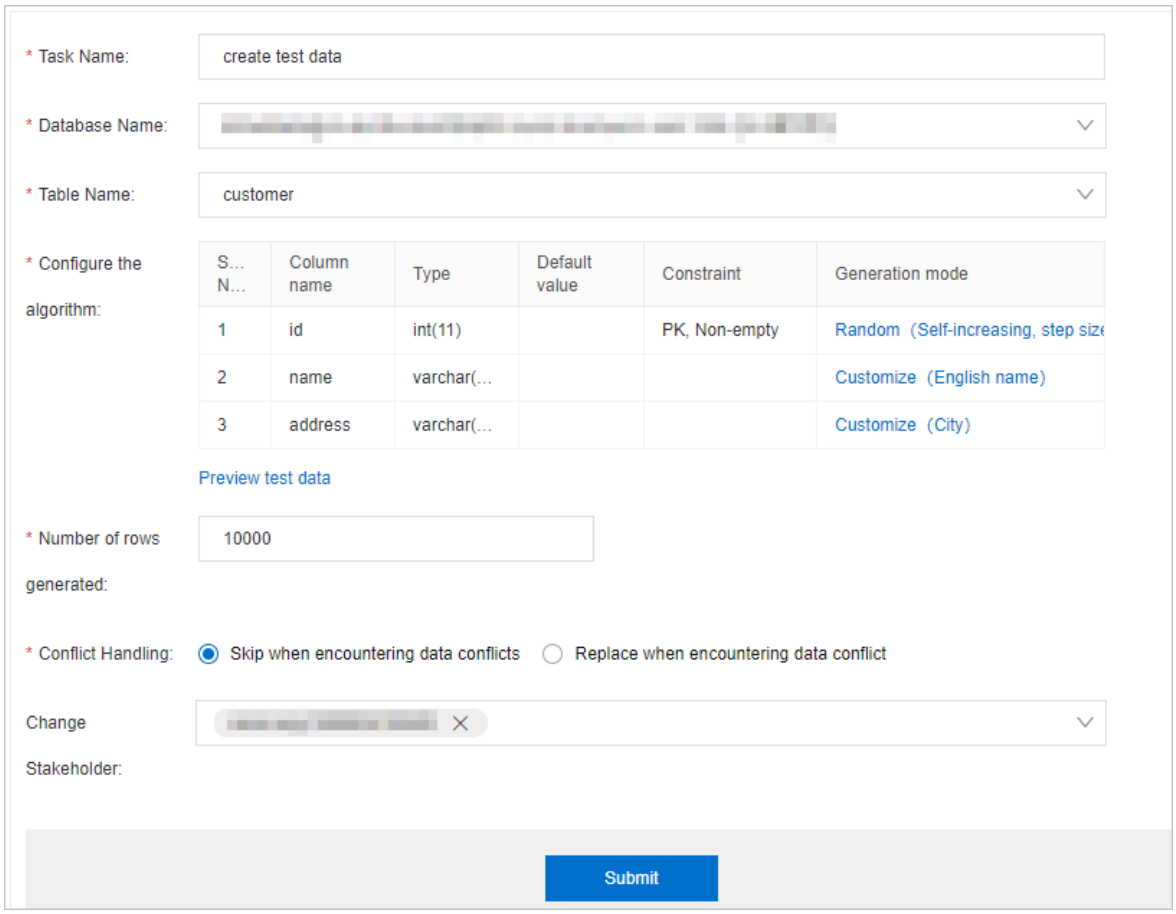
In view of this, DMS provides the Test Data Generate feature that allows you to generate test data in a quick, efficient, and secure manner. In addition, the discreteness of the data that is generated by using this feature is controllable.

Usage notes

- You can use this feature to generate test data for one table at a time. To generate test data for multiple tables, you must submit a ticket to generate test data for each table.
- To prevent database overload that is caused by instantaneous generation of excessive data, DMS allows you to perform traffic throttling. You can generate test data based on the following performance metrics:
  - One million rows of data with four fields can be generated in about 1 minute.
  - One million rows of data with 40 fields can be generated in about 2 to 3 minutes.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Data Plans > Test Data Generate**.
3. In the upper-right corner, click **Test Data Generate**.
4. In the Test data build ticket application dialog box, set the parameters as required.



Parameter	Description
Task Name	The name of the task. This facilitates management in subsequent operations.
Database Name	The database where the table for which you want to generate test data resides.
Table Name	The table for which you want to generate test data. Enter a keyword and select a table from the matched results. After you select the table, the **Configure the algorithm** parameter that contains the field information of the table is displayed. **Note** You can use this feature to generate test data for one table at a time. To generate test data for multiple tables, you must submit a ticket to generate test data for each table.

Parameter	Description
Configure the algorithm	The algorithms that you use to generate test data. Find the field for which you want to configure the algorithms, click the value of the Generation mode parameter, and then set the parameters in the Generation mode dialog box based on your business requirements. **Note** For example, you can use the Random, Customize, and Enumeration algorithms to generate test data of the STRING type. The Customize algorithm can be used to generate standard types of data for multiple industries.
Number of rows generated	The number of rows that you want to generate for the test data.
Conflict Handling	Specifies how DMS handles conflicts based on your business requirements.
Change Stakeholder	The stakeholders of the ticket. All specified stakeholders can view the ticket details and assist in the approval process. Irrelevant users other than DMS administrators and DBAs are not allowed to view the ticket details.

5. After you configure the settings, click **Submit**.
6. After the ticket is approved, DMS automatically starts to generate test data.

19.1.6.5. Clone databases

DMS provides the Database Clone feature that allows you to clone databases. This topic describes how to use this feature to clone databases.

Prerequisites

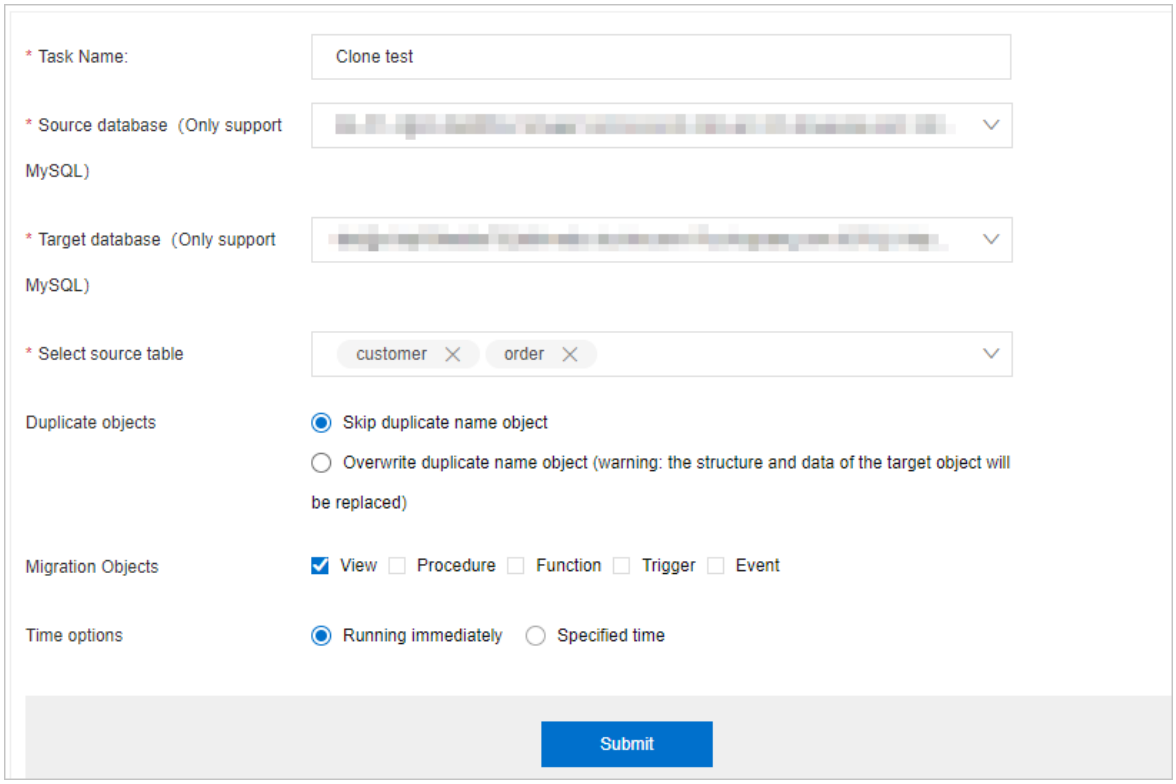
- A MySQL database is used.
- A database instance is managed in Flexible Management mode. You have logged on to the database instance in the DMS console.

Scenarios

- Create a full database backup.
- Initialize databases that are deployed across multiple environments, such as online and offline environments.
- Copy data from a database in an online environment to a database in an offline environment for data processing and analysis.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Data Plans > Database Clone**.
3. In the upper-right corner, click **Database Clone**.
4. In the Database clone order apply dialog box, set the parameters as required.



Parameter	Description
Task Name	The name of the task. This facilitates management in subsequent operations.
Source database (Only support MySQL)	The source database that you want to clone. You can enter a keyword to search for databases and select a database from the matched results.
Target database (Only support MySQL)	The destination database to which you want to write the data that is cloned from the source database. You can enter a keyword to search for databases and select a database from the matched results. **Note** The destination database and source database cannot be the same database.
Select source table	The one or more tables that you want to clone from the source database. You can enter a keyword to search for tables and select a table from the matched results. **Note** To clone all tables, set this parameter to All Tables.
Duplicate objects	The method that is used to handle object conflicts based on your business requirements. ◦ **Skip duplicate name object**: Skip the object that has a duplicate name. ◦ **Overwrite duplicate name object (warning: the structure and data of the target object will be replaced)**: Overwrite the schema and data of the object in the destination database with those in the source database.
Migration Objects	The objects that you want to clone. In addition to tables, you can synchronously clone objects such as views, stored procedures, functions, triggers, and events from the source database to the destination database.

Parameter	Description
Time options	Valid values: **Running immediately** and **Specified time**. If you set the Time options parameter to **Specified time**, you must specify the specific date and time for running the task. ○ **Running immediately**: The task is immediately run after the ticket is approved. ○ **Specified time**: DMS automatically runs the task to clone data at the specified time.

5. After you configure the settings, click **Submit**.
6. After the ticket is approved, the task is automatically run based on the specified time.

19.1.7. Data factory

19.1.7.1. Task orchestration

The task orchestration feature is powered by a distributed scheduling engine that is developed by Alibaba Cloud. You can use this feature to create task flows and schedule task flows to be run as needed. In addition, a variety of task nodes are provided. This way, you can use this feature to perform data archiving, data integration, and data processing as needed.

Scenarios

- Periodically archive and analyze business data.
- Synchronize online data to data warehouses for complex analysis.
- Periodically clean and process offline data.
- Orchestrate and periodically schedule tasks by performing database DDL or data manipulation language (DML) operations.

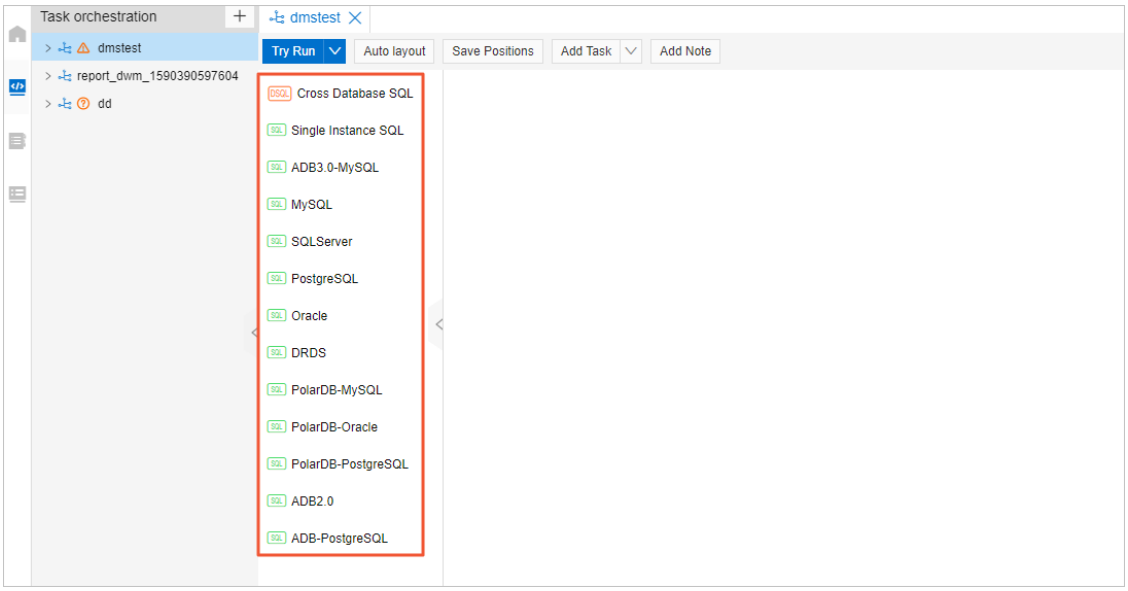
Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Data Factory > Task Orchestration**.
3. Click the  icon on the left-side navigation submenu.
4. Configure a task flow.
   i. Click the  icon next to **Task Orchestration**.
   ii. In the dialog box that appears, set the Task Flow Name and Description parameters.
   iii. Click **OK**.
5. Configure a specific task in the task flow.

   Note You can repeat this step to configure multiple tasks for the task flow.

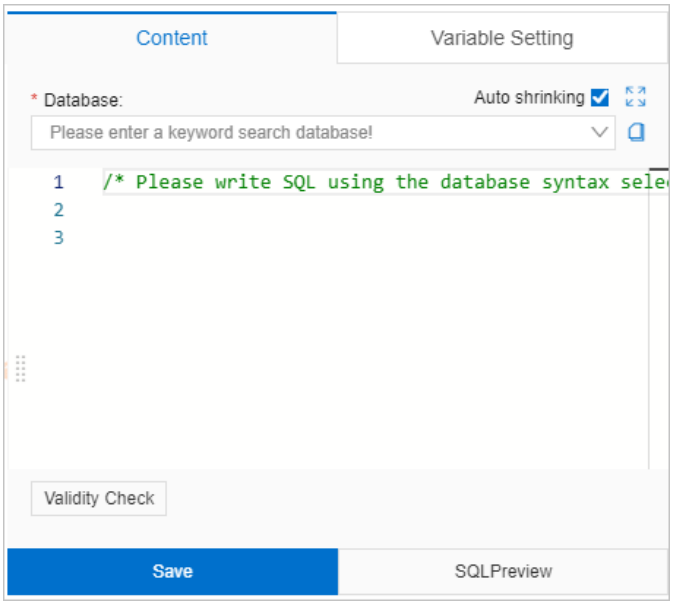
   i. In the left-side navigation pane of the Task Orchestration tab, find the task flow that you create and double-click the task flow name.

ii. On the Task Orchestration tab, drag a specific type of node from the task node list to the canvas where you can create a directed acyclic graph (DAG) based on your business requirements.



iii. In the DAG, click the task node that you want to configure.

iv. In the right-side pane, click a tab and set the parameters as required.

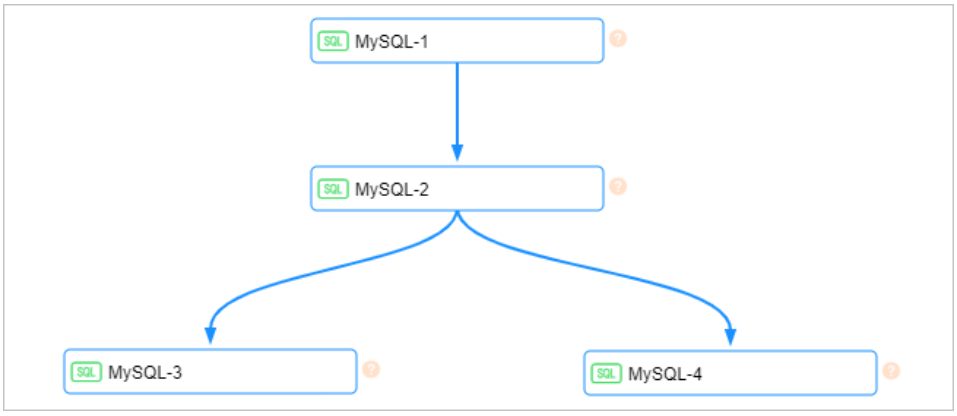


Tab	Parameter	Description

Tab	Parameter	Description
Content	**Database**	a. Select the database that you want to manage from the drop-down list. b. Enter the SQL statements to be executed in the field. c. Click **Save**. d. In the dialog box that appears, select **Existed table** or **New table** to store the query results. **Note** If you select **Existed table**, you must select a table name where you want to store the query results from the drop-down list. If you select **New table**, you need to enter a table name. e. Click **OK**.
Variable Setting	**Variable Name**	`bizdate` is the only default system variable, which indicates the previous day of the day when a task is run. The value of bizdate is in the yyyy-MM-dd format. If the default system variable cannot meet your business requirements, you can create a custom variable and enter the variable name in the Variable Name field. **Note** To delete a configured variable, click the icon.
	Variable Rule	To configure a variable rule, set the Time Format parameter and specify the operator, integer value, and time unit. Then, click **Save**. You can click **Increase Variable** to create more variables. **Note** After you configure a variable, you can reference the variable in the `${Variable name}` format in SQL statements on the **Content** tab. You can also click **SQLPreview** to check whether the variable is configured.

6. Configure node dependencies in the task flow.
   i. In the DAG, move the pointer over a task node and click and hold the circle, as shown in the following figure. Then, draw a line from the circle to the next node.

ii. Repeat the previous step to configure dependencies between task nodes based on your business requirements. This way, the sequence for running each task is established.



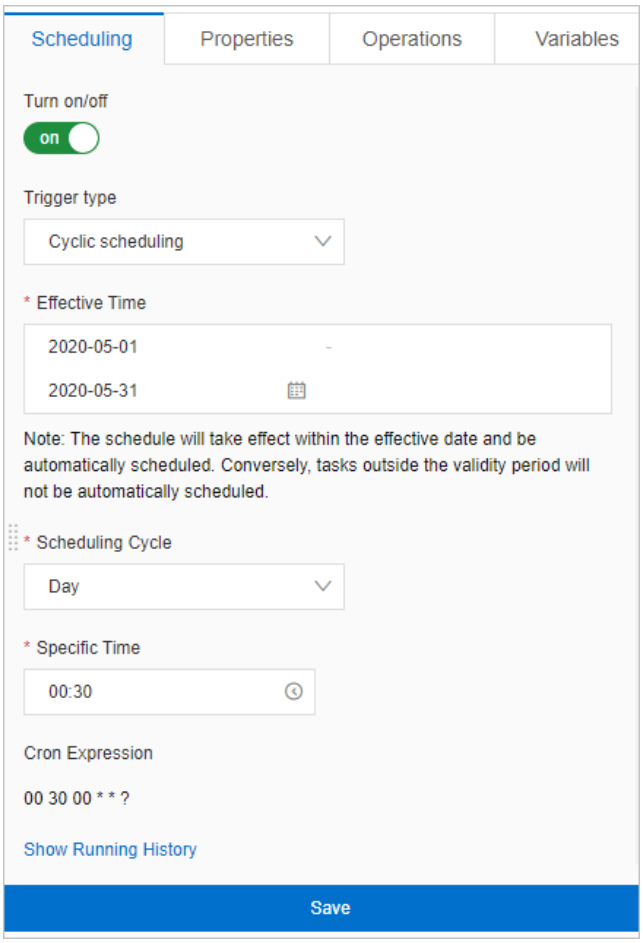
Note In the preceding figure, the MySQL-1 node is the first task to be run and the MySQL-2 node is the second. After that, the MySQL-3 and MySQL-4 nodes are run at the same time.

7. Configure scheduling information for the task flow.

i. Click the blank area on the canvas. In the right-side pane, click the **Scheduling** tab.

Note You can also click the **Properties** tab to configure the basic information of the task flow or click the **Operations** tab to view the operations that have been performed on the task flow.

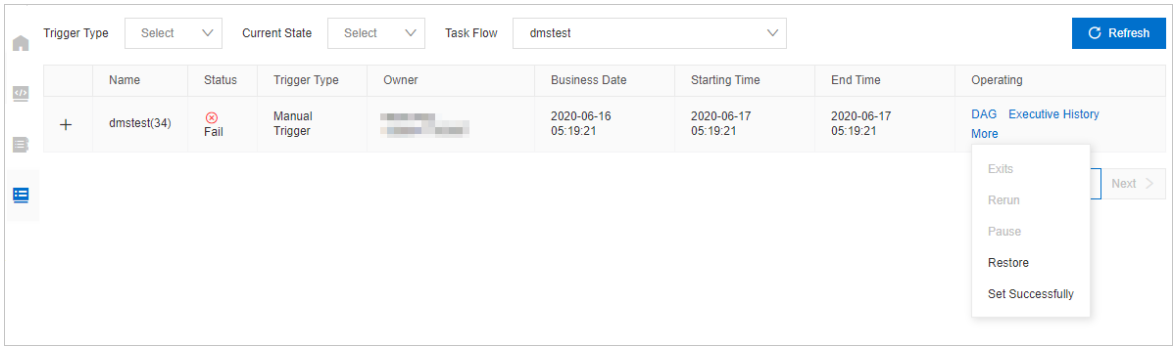
ii. Configure the scheduling information.



Parameter	Description
Turn on/off	Turn on **Turn on/off** to enable scheduling.
Trigger type	Set this parameter based on your business requirements. ■ When you set this parameter to **Cyclic scheduling**, you must set the **Effective Time**, **Scheduling cycle**, and **Specific Time** parameters. ■ When you set this parameter to **Schedule once**, you need only to set the **Specific Time** parameter.

iii. Click **Save**.

8. After you complete the configurations, click **Try Run** in the upper-left corner to check whether the task flow can be run as required.
9. Click the icon on the left-side navigation submenu, specify filter conditions, and then view the status of the task flow.



> **Note** Click the + icon before the task flow name to view the status of each task.

You can perform the following operations on the task flow:

- **DAG**: View the DAG of the task flow.
- **Executive History**: View the details of the operations that have been performed on the task flow.
- **More > Exits**: Stop running the task flow.
- **More > Rerun**: Rerun the task flow that is run or fails to be run.
- **More > Pause**: Pause the running of the task flow.
- **More > Restore**: Resume running the paused task flow.
- **More > Set Successfully** : Set the status of the task flow that fails to be run to Success.

19.1.7.2. Data warehouse development

19.1.7.2.1. Overview

Data Management (DMS) provides the data warehouse development feature. This feature uses databases as the computing engine and integrates a variety of tools and services in the database ecosystem. This allows you to develop and manage data warehouses with ease. This feature is designed to provide you with a one-stop development platform for data integration, processing, visualization, and value mining.

Benefits

- A variety of data warehouse engine types

  You can choose a data warehouse engine type based on your enterprise scale, data volume, and requirement for real-time performance. For example, you can choose AnalyticDB for MySQL or ApsaraDB RDS for MySQL as the data warehouse engine type.

- Two development modes

  The data warehouse development feature of DMS provides two development modes: task orchestration and professional development. The two modes meet different business requirements.

  - Task orchestration: This development mode allows you to develop a data warehouse by creating task flows and writing SQL scripts for task nodes. You do not need expertise in data warehouse development. You need only to focus on your business logic.
  - Professional development: This development mode meets the requirements of professional warehouse developers. It provides capabilities such as theme management, hierarchical management, production, release, multi-person collaboration, and data quality control. These capabilities empower professional warehouse development solutions for your enterprise.

> **Note** Some of the capabilities are planned to be supported soon.

- Support for offline and real-time data warehouses

  The data warehouse development feature supports offline data synchronization and task scheduling. This allows you to develop offline data warehouses with ease in DMS. In addition, DMS is integrated with Data Transmission Service (DTS) and cloud-native data warehouses. This allows you to build a real-time data warehouse system based on the real-time synchronization feature of DTS and cloud-native data warehouse engines. Then, you can develop data and consume data in real time in DMS.

- Unified management of online and offline data

  DMS supports unified database management and permission management. You can manage your online transaction processing (OLTP) databases and online analytical processing (OLAP) databases in a centralized manner in DMS. This avoids security issues that are caused by the isolation between offline and online systems.

19.1.7.2.2. Create a data warehouse project

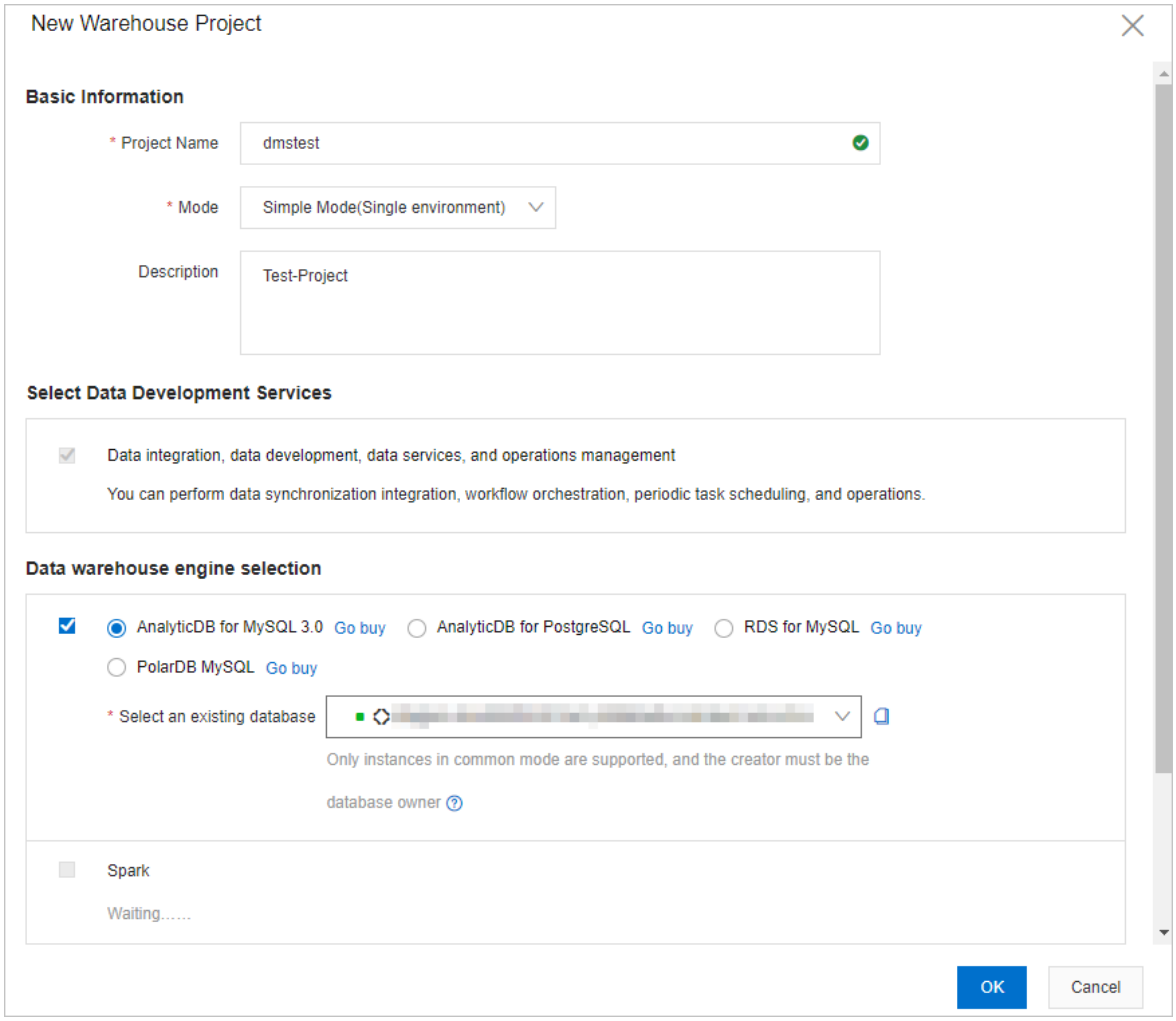
Before you can use the data warehouse development feature of Data Management (DMS), you must create a data warehouse project and select a data source for data warehouse development. This topic describes how to create a data warehouse project.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Data Factory > Data Warehouse Development**.
3. On the left-side navigation submenu, click the  icon.
4. Click the

   icon to the right of **Data warehouse**.
5. In the New Warehouse Project dialog box, set the parameters as required.



Section	Parameter or operation	Description
Basic Information	Project Name	The name of the project. Specify an informative name for easy identification.
	Mode	The mode of the project. Set this parameter to **Simple Mode(Single environment)**, which means that you can use the same database in a development environment and a production environment.
	Description	The description of the project.
Select Data Development Services	N/A	DMS automatically completes the configuration in this section.

Section	Parameter or operation	Description
Data warehouse engine selection	Select a data warehouse engine type by selecting the corresponding check box.	After you select a data warehouse engine type, such as **AnalyticDB for MySQL 3.0**, select a database from the **Select an existing database** drop-down list. **Note** Only databases of the instances that are managed in Security Collaboration mode are available in the drop-down list. You must be the owner of the selected database.

6. Click **OK**.

What's next

Create or import an internal table

19.1.7.2.3. Create or import an internal table

After you create a data warehouse project in Data Management (DMS), you must create or import an internal table for the project. An internal table refers to a table that exists in the data warehouse engine. This topic describes how to create or import an internal table.

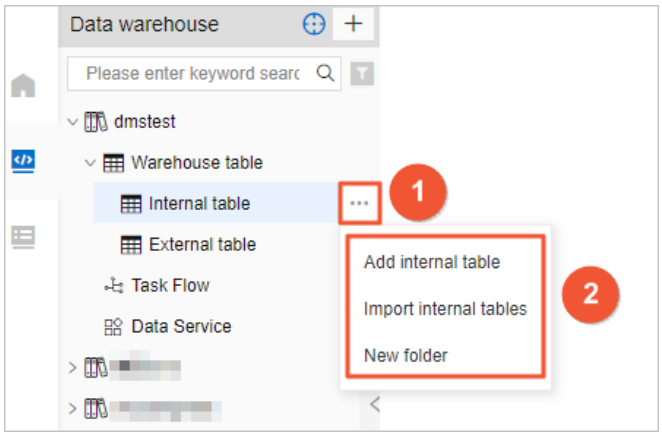
Prerequisites

You have the change permissions on the database for which you want to create or import an internal table. For information about how to apply for permissions, see Apply for permissions.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Data Factory > Data Warehouse Development**.
3. On the left-side navigation submenu, click the icon.
4. In the left-side navigation pane, expand a data warehouse project and then expand Warehouse table. Move the pointer over **Internal table** and then the More icon that appears. Then, select an option from the menu to perform one of the following operations.

   Note You cannot configure external tables in DMS.

   - Create an internal table:

a. Select **Add internal table**.
b. On the tab that appears, enter an SQL statement to create a table.
c. Click **Execute**.

- Import an internal table:

> **Note** The data warehouse development feature does not support real-time synchronization of tables that are created by using other means such as a command-line tool. You can import such tables to data warehouse projects in DMS.

a. Select **Import internal tables**.
b. In the Import internal tables dialog box, select the table that you want to import from the **Choose table** drop-down list and enter a description in the Remarks field.
c. Click **OK**.

- Create a folder:

> **Note** If you have a large number of tables, you can use folders to organize and classify the tables.

a. Select **New folder**.
b. In the New folder dialog box, enter a folder name.
c. Click **OK**.

What's next

Manage task flows

19.1.7.2.4. Manage task flows

Data Management (DMS) supports task flows and timed scheduling. You can configure a variety of task nodes in task flows. This can meet your requirements for data archiving, integration, and processing.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Data Factory > Data Warehouse Development**.
3. On the left-side navigation submenu, click the  icon.
4. In the left-side navigation pane, expand a data warehouse project. Move the pointer over **Task Flow** and then the
   
   icon that appears.
5. In the New Task Flow dialog box, enter a name and the description for the task flow.
6. Click **OK**.
7. On the tab that appears, configure task nodes for the task flow.

> **Note** The configurations of a task flow in professional development mode are basically the same as the configurations of a task flow in task orchestration mode. For more information, see Step 5 in the *Task orchestration* topic.

19.1.7.2.5. Use the data service feature

In Data Management (DMS), the data warehouse development feature is integrated with the data service feature. The data service feature allows you to export the data that is managed by DMS. This feature is applicable to scenarios where you need to export data at the column or row level, display data in a visualized manner, or perform complex analysis.

Limits

When you use the data service feature to create an API for a data warehouse, the data source of the API must be a table in the data warehouse project.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Data Factory > Data Warehouse Development**.

   > **Note** You can also choose **Data Factory > Data Service** in the top navigation bar.

3. On the left-side navigation submenu, click the icon.
4. In the left-side navigation pane, expand a data warehouse project and double-click **Data Service**.

Configure an API

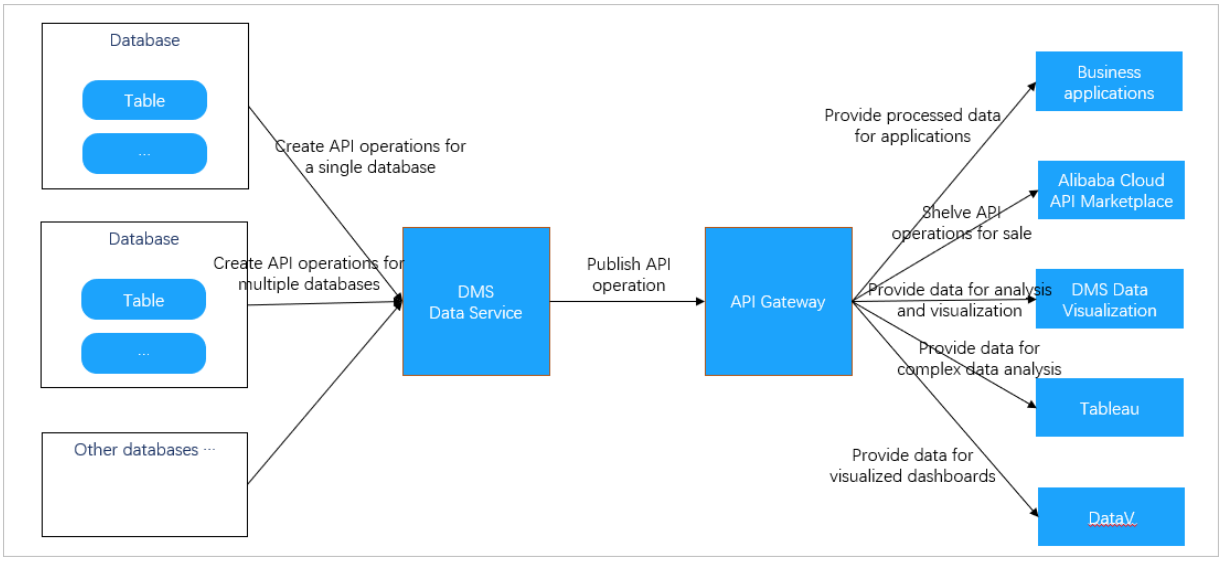
For information about how to configure an API, see Develop an API, Unpublish or test an API, Test an API, and Call an API.

19.1.7.3. Data service

19.1.7.3.1. Overview

Data Management (DMS) provides the data service feature, which allows you to export the data that is managed by DMS. This feature is applicable to scenarios where you need to export data at the column or row level, display data in a visualized manner, or perform complex analysis.

Features



- You can use the data service feature to create APIs that can be called to access the data that is managed by DMS. When you create the APIs, you can apply the security control features that are used for SQL execution in the SQLConsole, such as permission control and data de-identification.
- The data service feature works based on a serverless architecture. This feature frees you from the concern about the infrastructure of the runtime environment, such as servers and networks. You need to focus only on how to create APIs and design data query logic. This avoids operations and maintenance (O&M) overheads that are generated by using traditional architectures.
- The data service feature is fully integrated with API Gateway. You can use this feature to publish APIs to API Gateway. This way, you can use all the features that are provided by API Gateway, such as API permission control, IP address-based access control, throttling, metering and billing, and SDKs.

Scenarios

Scenario	Description
Minimize data exposure	Assume that you need to export the data that is managed by DMS to an external environment. In this case, APIs can be called to export the data of specific rows or columns to the external environment. To export the data of specific rows, specify a filter condition in the SQL statement. To export the data of specific columns, specify the columns in the SQL statement. Compared with data export of a whole table, this minimizes data exposure and ensures data security.
Connect visualization tools to databases	Most visualization tools can connect to databases by calling APIs. You can connect a visualization tool to your database by calling an API, instead of by using a username and a password. This method is easy to implement and avoids account exposure.
Sell APIs in the Alibaba Cloud Marketplace	If you want to provide paid or free data for other users, publish an API to the Alibaba Cloud Marketplace.
Provide processed data for applications	After data is processed and summarized by using the data warehouse development feature of DMS, APIs can be created and provided for applications to read the processed data from DMS to meet business needs. To modify the logic of data reading, you need only to modify the query logic of the required API without the need to republish the application.

19.1.7.3.2. Develop an API

The data service feature of Data Management (DMS) provides comprehensive capabilities to help you develop APIs with ease. This topic describes how to create and manage APIs.

Prerequisites

API Gateway is activated. For more information, see the documentation of *API Gateway*.

Context

The data service feature allows you to export the data that is managed by DMS. This feature is applicable to scenarios where you need to export data at the column or row level, display data in a visualized manner, or perform complex analysis. For more information, see Overview.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Data Factory > Data Service**.
3. Click the **API Development** tab on the left side.
4. On the APIManagement tab, click **New API** in the upper-right corner.
5. On the tab that appears, set the parameters as required.

i. Set the parameters on the AttributeConfiguration tab.

Parameter	Description
APIName	The name of the API. The name must be 4 to 100 characters in length and can contain letters, digits, and underscores (_). The name must start with a letter.
Description	Optional. The description of the API. Enter an informative description such as a description of the data to be returned by the API or the scenarios in which the API can be called.
Path	The path of the API. The path must start with a forward slash (/) and can contain letters, digits, underscores (-), and hyphens (-). The specified path forms a part of the URL that is used to call the API. A URL that is used to call an API must be in the `https://{Domain name}{Path}` format. For example, if the domain name is `xxxx-cn-hangzhou.alicloudapi.com` and the path is specified as `/item/monthly_data`, the URL that is used to call the API is `https://xxxx-cn-hangzhou.alicloudapi.com/item/monthly_data`.
ReturnFormat	The format of the data to be returned by the API. Set the value to **JSON**.
RequestMode	The method to be used to call the API. Valid values: **POST** and **GET**.
TimeOut (MS)	The maximum length of time during which a response to an API request must be received from the backend service of the called API. Unit: milliseconds. If the response time exceeds the specified time, API Gateway returns a timeout error. The maximum response time is 30,000 milliseconds, which equals 30 seconds.
Returns the maximum number of records	The maximum number of entries that can be returned for an API request. This parameter limits the number of entries that can be returned for each query that is performed after the API is called. **Note** If the database instance is managed in Security Collaboration mode, the value of this parameter must be smaller than the maximum number of entries that is specified in the security rules.

ii. Click the **ExecuteConfiguration** tab and set the parameters on this tab.

Parameter	Description
Instance query type	The type of instance query. Valid values: ■ **Single InstanceQuery**: The API can be called to read data from only one database instance. ■ **Cross-instanceQuery**: You can write dynamic SQL statements for the API to query data across multiple database instances. **Note** After you set this parameter to **Cross-instanceQuery**, you need only to enter dynamic SQL statements in the **QuerySQL** field.
Data source	The database to be queried when the API is called. You can enter a keyword to search for the databases on which you have query permissions and then select a database.

Parameter	Description
ConfigurationMode	The method that is used to configure data query. Valid values: ■ **Table boot mode**: You can configure data query by selecting a table and fields. ■ **Script mode**: You must configure data query by defining variables and writing SQL statements. **Note** After you set this parameter to **Script mode**, you need only to enter SQL statements in the **QuerySQL** field.
SelectTable	The table to be queried. You can enter a keyword to search for a table.
FieldList	The fields in the selected table. You can specify the fields as request parameters or response parameters as needed.
Script mode	The mode in which an SQL script is written to define the data query logic. **Note** You can set the **ConfigurationMode** parameter to **Table boot mode** or Script mode.
QuerySQL	The SQL statement that is used to query the data in the table. After you enter the SQL statement, click **ParsingScript** to verify whether the syntax is valid and parse the request parameters and response parameters. **Note** ■ Custom variables are supported. Custom variables can be mapped as request parameters in API requests. The variables in the SQL statement must be defined in the `${Variable name}` format. For example, you can define the `${category}` variable and use it in the following SQL statement: `select item_id, item_name from ex_item where category=${category}` . ■ If you set the **Instance query type** parameter to **Cross-instanceQuery**, you must use the syntax of cross-database query SQL statements. For more information, see Cross-database query.

iii. Click the **Request Parameters** tab and set the parameters on this tab.

Parameter	Description
ParametersName	The name of the request parameter. ■ The name can contain letters, digits, hyphens (-), and underscores (_). ■ The name must start with a letter or an underscore (_). ■ The name must be 1 to 50 characters in length.
FieldName	The name of the field referenced by the request parameter. The field name is specified on the **ExecuteConfiguration** tab and cannot be changed.
Cannot be empty	Specifies whether the request parameter is required.
Description	The description of the request parameter.
Data type	The data type of the request parameter. The data type is used to verify whether the value of the request parameter in an API request is valid. Valid values: String, Integer, and Floating point. Default value: String. **Note** This parameter affects the SQL statement to be executed when the API is called.
Example value	The sample value of the request parameter. Sample values that are provided in SDKs and documentation can help users call APIs.
Default value	The default value of the request parameter. If the request parameter is optional and not specified in the API request, the default value is used.

iv. Click the **Return parameter** tab and set the parameters on this tab.

Parameter	Description
ParametersName	The name of the response parameter. ■ The name can contain letters, digits, hyphens (-), and underscores (_). ■ The name must start with a letter or an underscore (_). ■ The name must be 1 to 50 characters in length.
FieldName	The name of the field referenced by the response parameter. The field name cannot be changed.
Description	The description of the response parameter.
Data type	The data type of the response parameter. Valid values: String, Integer, and Floating point. Default value: String. This parameter is used by DMS to convert the type of the data in API responses. This parameter affects the JSON data that is returned after the API is called.
Example value	The sample value of the response parameter. Sample values that are provided in SDKs and documentation can help users understand API responses.

6. Click **Save**.
7. Click the **API Development** tab on the left side.
8. Perform the following operations to manage the API based on your business requirements:

- Publish the API:

  On the APIManagement tab, find the API and click **Publish** in the **Operation** column. In the message that appears, click **OK**.

- Modify the API:

  On the APIManagement tab, find the API and click **Modify** in the **Operation** column. Modify the configurations of the API based on the description in Step 5 and click **Save**.

- Delete the API:

  On the APIManagement tab, find the API and click **Delete** in the **Operation** column. In the message that appears, click **OK**.

19.1.7.3.3. Unpublish or test an API

This topic describes how to unpublish or test an API that has been published.

Prerequisites

An API is created. For more information, see Develop an API.

Procedure

1. Log on to the DMS console.
2.
3. Click the **API Publish** tab on the left side.
   The APIPublishList tab displays all the published APIs.
4. Find the API that you want to manage and perform the following operations based on your business requirements:
   - Unpublish the API:

     Click **Offline** in the **Operation** column. In the message that appears, click **OK**.

   - Test the API:

     Click **Test** in the **Operation** column. For more information, see Test an API.

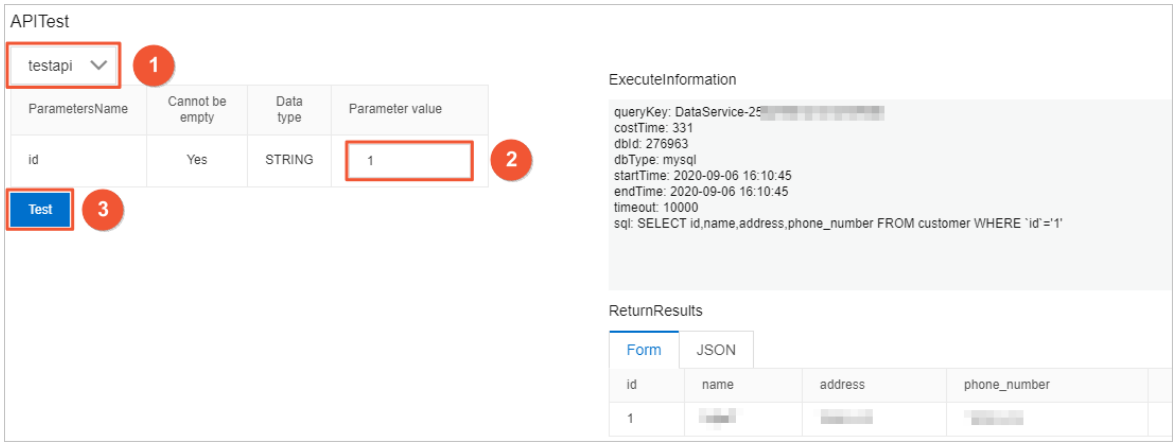
19.1.7.3.4. Test an API

After you create an API, you can test the API to verify whether the API meets your business requirements.

Prerequisites

Procedure

1. Log on to the DMS console.
2.
3. Click the **API Test** tab on the left side.
4. On the APITest tab, test an API.



i. Select the API that you want to test from the drop-down list.

ii. Enter values in the Parameter value column.

iii. Click **Test**.
After the test is complete, the execution information and return results appear on the right side. You can evaluate whether the API meets your business requirements based on the information.

> **Note** You can click the **JSON** tab in the **ReturnResults** section so that the return results are displayed in the JSON format.

19.1.7.3.5. Call an API

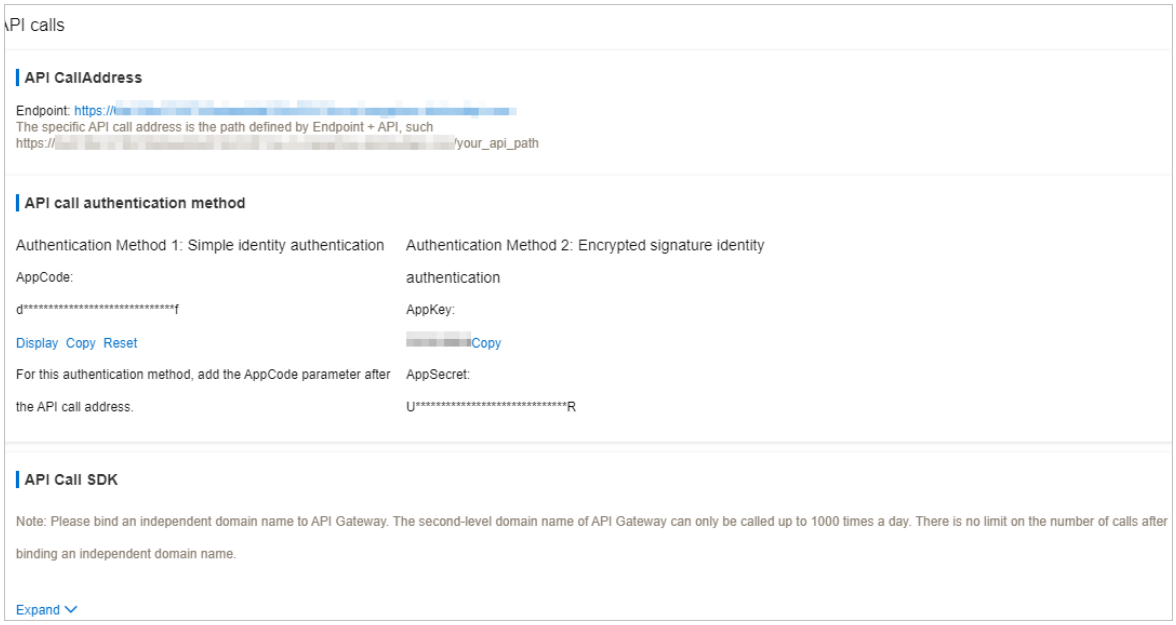
After you create, publish, and test an API, you can call the API in an application by using an SDK.

Prerequisites

- An API is created and published. For more information, see Develop an API.
- API Gateway is activated. For more information, see the documentation of *API Gateway*.

Procedure

1. Log on to the DMS console.
2.
3. Click the **API Call** tab on the left side.
4. View the API call address and the authentication information.



- **Simple identity authentication**: requires only an AppCode. This authentication method is suitable for calling APIs by using URLs. This authentication method has a low security level and is generally used in scenarios in which data visualization is involved, such as calling APIs in DataV.
- **Encrypted signature identity authentication**: requires an AppKey and an AppSecret, which are used to dynamically generate an encrypted signature for calling an API. This authentication method has a high security level.

5. Call the API in an application by using an SDK.

> **Note** For more information about how to call an API in an application by using an SDK, see the documentation of *API Gateway*.

19.1.8. Schemas

19.1.8.1. Design a schema

Data Management (DMS) provides the schema design feature. This feature allows you to change schemas with ease. This topic describes how to use this feature.

Prerequisites

A MySQL, a PolarDB-X, or an ApsaraDB for OceanBase database is available.

Context

When you create projects, process new business requirements, or optimize business operations, you may need to create tables or change schemas. For example, you may need to add or delete fields or indexes, adjust field attributes, or adjust the index composition. In these scenarios, you can use the schema design feature of DMS.

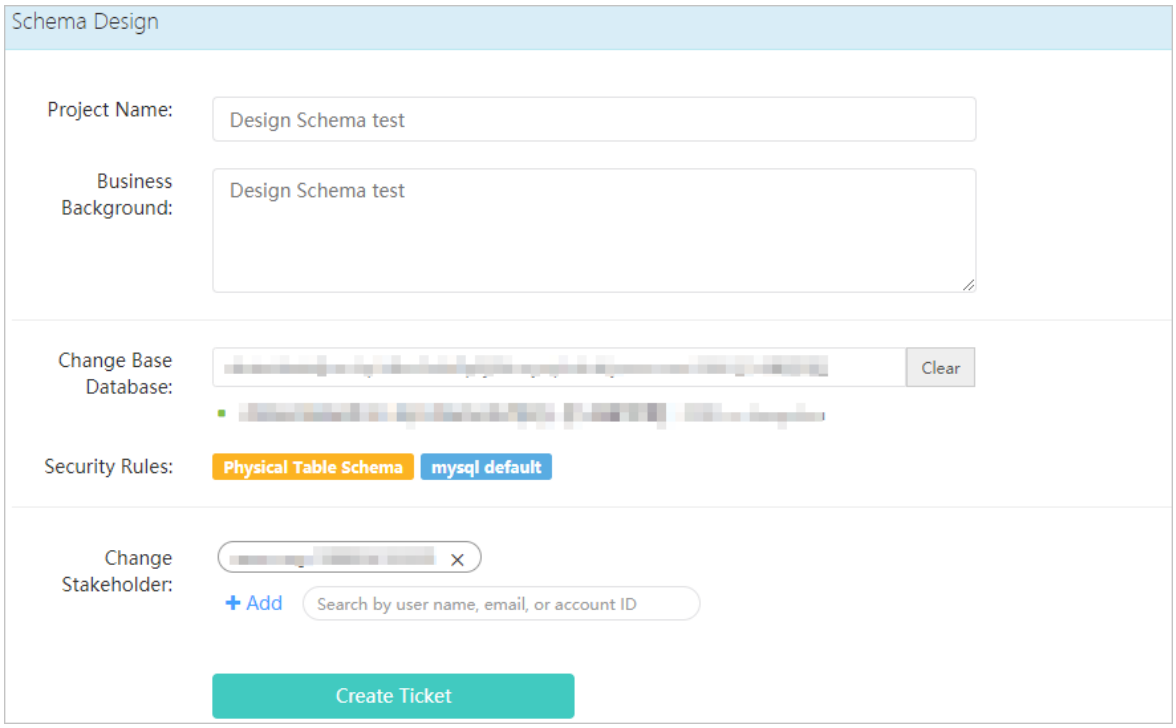
- This feature allows multiple users to collaboratively design schemas on web pages.
- This feature allows you to send verified scripts to other environments with ease. This ensures consistency between schemas in different environments.

Usage notes

When you submit a schema design ticket to delete a table, make sure that the table was created by submitting a schema design ticket.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Schemas > Schema Design**.
3. On the Schema Design tab, click **Schema Design** in the upper-right corner.
4. On the Schema Design tab, set the parameters for a schema design ticket.



Parameter	Description
Project Name	The name of the project. Set an informative name for easy identification.
Business Background	The purpose or objective of the project, which accelerates the approval process.
Change Base Database	The database whose schema you want to change. You can enter a prefix to search for a database. Only databases on which you have permissions in test or development environments are available. **Note** You must have at least one of the query, export, and change permissions on the selected database.
Security Rules	The security rules that you want to apply. DMS automatically selects security rules based on the selected database.
Change Stakeholder	The stakeholders of the changes. The specified stakeholders can view the ticket details and assist in the approval process. Irrelevant users other than DMS administrators and database administrators (DBAs) are not allowed to view the ticket details.

5. Click **Create Ticket**.
6. Design a schema based on your business requirements.
   - Create a table:

a. Click **Create Physical Table**.

> **Note** If the selected database is a logical database, click **Create Logical Table**.

b. On the Create Physical Table tab, configure information such as the table name, character set, fields, and indexes.

c. Click **Save**.

> **Note** After you click Save, DMS prechecks the configured information based on design specifications. If the information does not conform to the design specifications, a prompt appears.

d. After the configured information passes the precheck, click **Confirm Changes and Submit to Save**.

- Modify the schema of a table:

a. In the left-side pane, click the table whose schema you want to modify.

b. On the menu that appears, select **Design Table**.

c. Modify the schema as needed and click **Save**.

> **Note** After you click Save, DMS prechecks the configured information based on design specifications. If the information does not conform to the design specifications, a prompt appears.

d. After the configured information passes the precheck, click **Confirm Changes and Submit to Save**.

7. After you complete the schema design, click **Perform Changes to Base Database**.
8. In the Perform Changes to Base Database dialog box, set the Execution Strategy parameter to **Execute Now** or **Schedule**.
9. Click **Submit for Execution** and wait until the ticket is approved.
10. After the ticket is approved, click **Go to Next Node**.

> **Note**
> - After the ticket is approved, DMS performs the changes at the specified time. If you do not specify the execution time, the changes are automatically performed after the ticket passes the last approval node. You can view the execution status and operations logs. After all changes are performed, you can repeat the preceding procedure to modify the design. If you are sure that the design is complete, click Go to Next Node.
> - The preset design specifications determine whether you can go back to the previous node and modify the design after the ticket is submitted to the next node.

11. In the Go to Next Node message, click **Go to Next Node**.
12. On the Project Homepage tab, click **Perform Changes to Target Database**.
13. In the Perform Changes to Target Database dialog box, set the Target Database and Execution Strategy parameters and click **Submit for Execution**.

> **Note** The database to which changes are performed must be in a production environment.

14. Wait until the ticket is approved and the changes are performed.
15. Click **Go to Next Node**.
    The schema design process ends and the ticket is closed.

19.1.8.2. Synchronize schemas

Data Management (DMS) provides the schema synchronization feature. You can use this feature to compare the schemas of two databases, generate a script for schema synchronization, and then run the script to synchronize schemas for the destination database. This topic describes the schema synchronization feature and shows you how to synchronize schemas.

Prerequisites

ApsaraDB for OceanBase or MySQL databases are available.

Usage notes

- You cannot synchronize schemas for databases in production environments.
- You can initialize an empty database based on part or all of the tables in a physical or logical database.

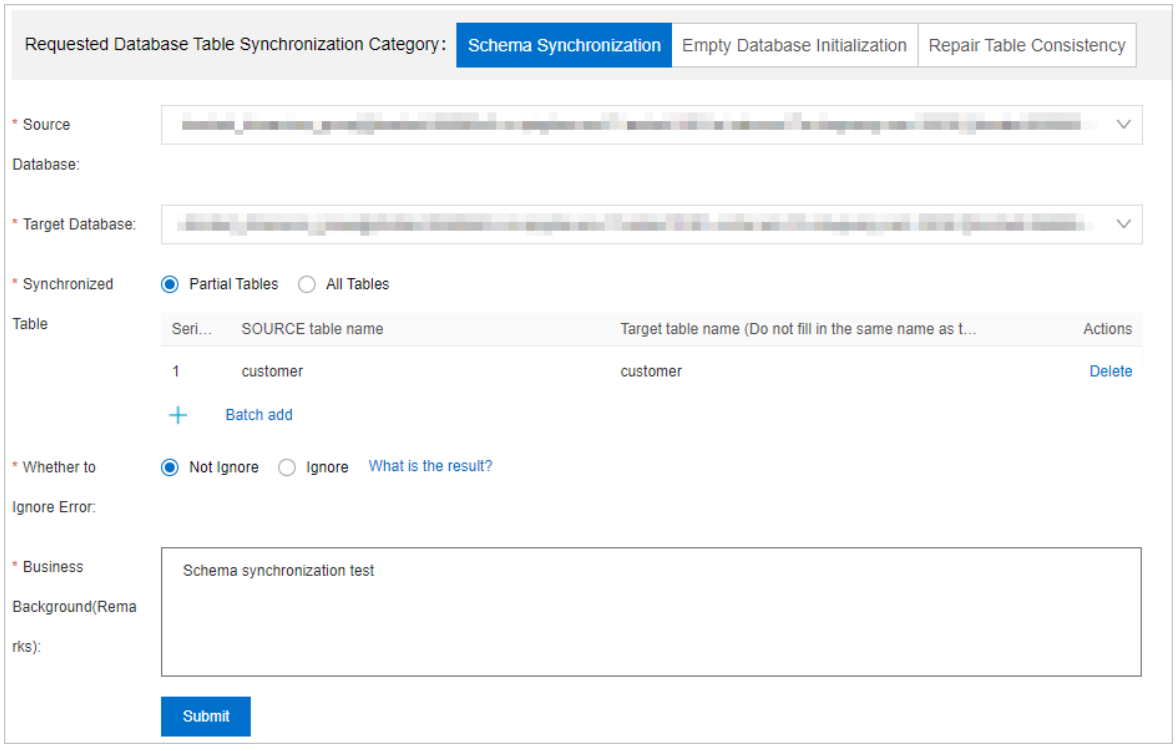
Scenarios

The schema synchronization feature can be used to synchronize schemas in the following scenarios to ensure schema consistency:

- Between a database in a production environment and a database in a test environment.
- Between databases in a test environment.
- Between databases in a production environment.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Schemas > Table Sync > Schema Synchronization**.
3. On the Schema Synchronization tab, set the parameters for a schema synchronization ticket.



Parameter	Description
Source Database	The name of the source database for schema synchronization. You must have query permissions on the source database.

Parameter	Description
Target Database	The name of the destination database for schema synchronization. You must have change permissions on the destination database.
Synchronized Table	The tables that you want to synchronize. Valid values: ○ **Partial Tables**: Synchronize some of the tables in the source database. You can click the plus icon to add tables one by one. Note If you do not specify the names of destination tables, DMS assumes that the destination table names are the same as the source table names. ○ **All Tables**: Synchronize all tables in the source database.
Whether to Ignore Error	Specifies whether to skip errors when SQL statements are being executed. Valid values: ○ **Not Ignore**: If an error occurs when SQL statements are being executed, DMS stops executing the current and subsequent SQL statements. ○ **Ignore**: If an error occurs when SQL statements are being executed, DMS skips the error and continues to execute subsequent SQL statements until all remaining statements are executed.

4. Click **Submit**. DMS starts to analyze the schemas.
5. Check the comparison results.

   Note If the schemas are changed during schema analysis, click Re-analyze in the Schema Analysis step.

6. Verify the script for schema synchronization and click **Submit and Synchronize to Target Database**.

   Note If the schemas of the source database and destination database are the same, you do not need to submit the script. The schema synchronization ticket is closed.

19.1.8.3. Initialize empty databases

DMS provides the empty database initialization feature. This feature allows you to compare the schemas of two databases, generate statements for synchronizing data from the source database to the destination database, and execute these statements against the destination database. The destination databases must be empty databases. This topic describes how to initialize empty databases.

Prerequisites

- The source databases and destination databases are MySQL or ApsaraDB for OceanBase databases.
- The destination databases are empty databases that contain no table.

Usage notes

The empty database initialization feature allows you to synchronize specific or all of the tables from a database, regardless of whether the database is a physical or logical database.

Scenarios

The empty database initialization feature is used to synchronize data between databases that are deployed in different regions or units. For example, this feature is applicable to the following scenarios:

- Synchronize data between one database that is deployed in the online environment and another in the offline environment.
- Synchronize data between different databases that are deployed in the offline environment.
- Synchronize data between different databases that are deployed in the online environment.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Schemas > Table Sync > Empty Database Initialization**.
3. On the Table/Database Synchronization Application tab, set the parameters as required to configure an Empty Database Initialization ticket.

Parameter	Description
Source Database	The name of the source database from which data is to be synchronized. You must have the read permissions on the source database.
Target Database	The name of the destination database to which data is to be synchronized. You must have the write permissions on the destination database. **Note** The type of the destination database must be the same as that of the source database.
Initialized Table	The tables that you want to synchronize. Valid values: ○ **Partial Tables**: Synchronize one or more tables in the source database. To add a table, click the **+** icon and specify a source table name. ○ **All Tables**: Synchronize all tables in the source database.
Whether to Ignore Error	○ **Not Ignore**: If an error occurs when SQL statements are being executed in serial mode, DMS immediately stops executing the current SQL statement and the remaining SQL statements. ○ **Ignore**: If an error occurs when DMS is executing an SQL statement, DMS stops executing the current SQL statement and continue to execute the next statement until all remaining SQL statements are executed.

4.
5.
6.

19.1.8.4. Repair table consistency

DMS provides the table consistency repair feature. This feature allows you to compare the schemas of tables in databases that are deployed in different environments, identify schema differences with efficiency, and execute SQL statements in the environment in which you want to modify the schemas. This ensures schema consistency in different environments.

Prerequisites

The source databases and destination databases are MySQL or ApsaraDB for OceanBase databases.

Scenarios

- Ensure the schema consistency between physical tables that are deployed in the offline environment and the online environment.

- Ensure the schema consistency between physical tables in a physical database and logical tables in a logical database.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Schemas > Table Sync > Repair Table Consistency**.
3. On the Table/Database Synchronization Application tab, set the parameters as required to configure a Repair Table Consistency ticket.

Parameter	Description
Base Database(Physical Database)	The source database based on which schema consistency is to be repaired. You must have the read permissions on the source database.
Target Database	The destination database whose data is to be modified. You must have the write permissions on the destination database.
Repaired Table	The pair of tables between which schema consistency is to be repaired. To add a pair of tables, click the **+** icon and specify the table names. **Note** If you do not specify the destination table name, DMS names the destination table after the name of the specified source table.
Whether to Ignore Error	○ **Not Ignore**: If an error occurs when SQL statements are being executed in serial mode, DMS immediately stops executing the current SQL statement and the remaining SQL statements. ○ **Ignore**: If an error occurs when DMS is executing an SQL statement, DMS stops executing the current SQL statement and continue to execute the next statement until all remaining SQL statements are executed.
Business Background(Remarks)	The business background of the ticket. This reduces unnecessary communication.

4.
5.
6.

19.1.9. SQL review

DMS provides the SQL review feature to help you prevent SQL statements that do not use indexes or do not conform to database development standards. This reduces the risk of SQL injection attacks.

Prerequisites

The environment type of the database instance in which you want to use the SQL review feature is **Test** in the DMS console. This is because SQL review is performed before DMS publishes SQL statements to an online environment.

Context

When you develop a project, you need to execute SQL statements on databases to add, delete, modify, and query data so that you can implement business logic and display data. Before the project is published, you must review all SQL statements that are used. This prevents SQL statements that do not conform to database development standards from being published to an online environment and accordingly impeding the business.

If DBAs manually review all SQL statements one by one, excessive human resources are consumed and the R&D efficiency is low. The SQL review feature helps you review SQL statements and also provides optimization suggestions.

Usage notes

- Only XML or TXT files can be uploaded.
- Tables that are involved in SQL statements must exist in the database that you select. Otherwise, DMS cannot review these SQL statements.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **Optimization > SQL Review**.
3. On the tab that appears, click **New** in the upper-right corner.
4. Configure information about an SQL review ticket.

Parameter	Description
Project name	Enter a project name based on your business requirements so that the ticket can be distinguished from others in subsequent processing.
Data source	Select the database in the test environment that is used in your project. You must have the change permission on the database.
Business Description	Enter detailed information about the business scope of the project as required to help relevant users know about the project.
Relevant personnel	Enter an at sign (@) and select a user. **Note** You can repeat this operation to select multiple users.
Upload a file	Click **Add**, select files, and then click **Upload**. **Note** ◦ The iBATIS and MyBatis files are in the XML format. ◦ SQL statements are saved as TXT files. Multiple SQL statements are separated by semicolons (;). ◦ To remove an added file, you can select the check box before the file name and click **Delete**.

5. Click **Submit application**.
6. View SQL review results.

> **Note**
>
> - If SQL statements in a file conform to database development standards and use indexes, DMS determines that these SQL statements pass the SQL review and offers no suggestion about indexes.
> - If SQL statements in a file conform to database development standards but do not use indexes, DMS determines that these SQL statements pass the SQL review and offers suggestions about indexes.
> - If SQL statements in a file do not conform to database development standards, DMS determines that these SQL statements fail the SQL review.

7. Find a failed SQL review result and click **View reason** to check the reason. You can also click **Details**, **Adjust SQL**, or **More** in the **Operation** column to perform other operations.

   > **Note** After you optimize SQL statements in a file and click **Confirm**, DMS reviews the SQL statements again. For dynamic SQL statements in XML files, you must optimize each SQL statement combination.

8. When all SQL statements pass the SQL review, click **Inspection results**.
9. In the dialog box that appears, click **Submit for approval** and wait for approval.

   > **Note** The approval process of the ticket is based on the security rules that are configured for the current database instance.

19.1.10. SQLConsole

19.1.10.1. Single database query

DMS provides the single database query feature that allows you to write SQL statements to query data. This feature is applicable to scenarios where you need to verify business code before it is published, analyze product effects, or identify issues in an online environment.

Prerequisites

You have the query permission on the database or table that you want to query.

Usage notes

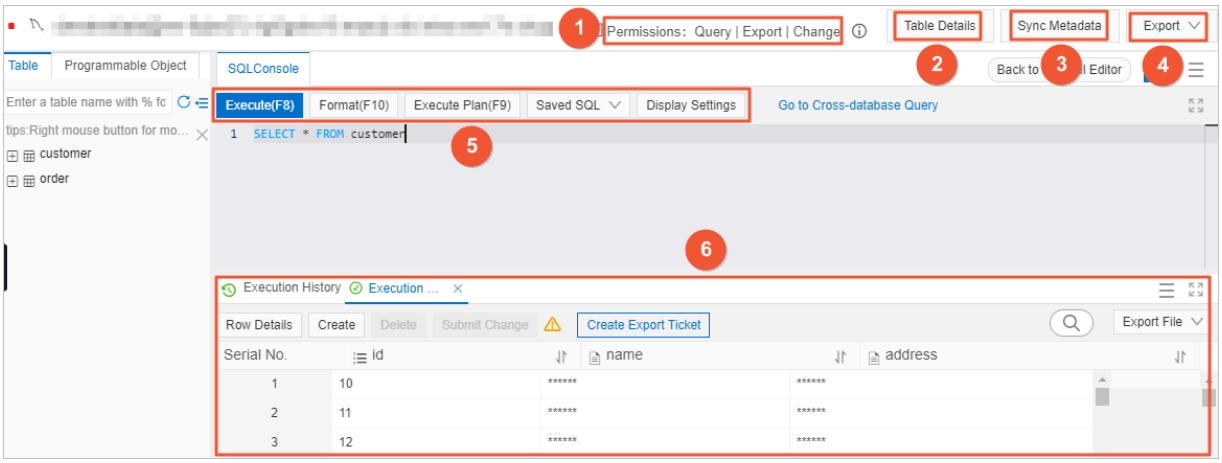
- If a table contains sensitive or confidential fields and you do not have permissions on the fields, you cannot use the fields in a WHERE clause to construct query conditions.
- If a table contains sensitive or confidential fields and you do not have permissions on the fields, the values of the fields are displayed as `******` in query results.
- In the left-side pane of the SQLConsole, you can search for tables in the current database. You can also view information about each table, such as the schema, fields, and indexes of the table. The pane can display 1,000 tables at most.
- By default, a maximum of 200 data rows can be returned for each query. If you are a DMS administrator, you can choose **System Management** > **Security** > **Security Rules** in the DMS console and modify this default number as needed.
- By default, a maximum of 10 GB of data can be queried each time. If you are a DMS administrator, you can choose **System Management** > **Security** > **Security Rules** in the DMS console and modify this default volume as needed.
- By default, the timeout period for executing each SQL statement is 60 seconds. If you are a DMS administrator,

you can find the database instance that you want to edit in the DMS console, open the **Edit instance** dialog box, and then modify the default timeout period as needed in the **Advanced information** section.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **SQLConsole > Single Database query**.
3. In the dialog box that appears, select a database from the drop-down list, or enter the keyword of a database name to search for databases and select the database that you want to query. Then, click **Confirm**.
4. In the SQLConsole, write an SQL statement and click **Execute**.
   After the SQL statement is executed, the execution results are displayed in the execution result section in the lower part.

Overview of the Single Database query tab



No.	Description
①	You can view the permissions that you have on the current database. You can move the pointer over the icon to view information about the database, such as the owner and the DBA of the database.
②	You can click the **Tables** icon to switch to the **Table List** tab, and click the **Query** icon to return to the SQLConsole.
③	You can click the **Sync Metadata** icon, and then click **OK** in the message that appears. DMS starts to collect the up-to-date metadata of the current database. The metadata includes the information about tables, fields, indexes, and programmable objects. The collected metadata is used to manage permissions on the tables, fields, and programmable objects in a fine-grained manner.
④	You can use the export feature to export data or schemas in the current database to a Word, Excel, or PDF file. You can also export the SQL statements that are used to create tables in the database.
⑤	You can write and execute SQL statements in the SQLConsole to query data in the current database. DMS also provides other features, such as Format, Execute Plan, Saved SQL, and Display Settings.

No.	Description
⑥	After SQL statements are executed, you can view the execution results. You can also perform operations on the result data. For example, you can view row details, or add, delete, or modify data. **Note** You can click the **Execution History** tab. On this tab, you can view the execution history of SQL statements. Each record includes information such as the start time, database, and executed SQL statement. You can also export the result data.

19.1.10.2. Cross-database query

DMS provides the cross-database query feature that allows you to perform join queries across online heterogeneous data sources that are deployed in different environments. Based on the cross-database query feature, you can perform join queries with ease across databases and tables in database instances that are registered in DMS.

Prerequisites

- The type of each database instance that you want to query is MySQL, SQL Server, PostgreSQL, PolarDB-X, or Redis.
- The cross-database query feature is enabled for each database instance.

  Note If the cross-database query feature is not enabled for a database instance, you can open the Edit instance dialog box in the DMS console, and then enable the feature and customize a database link name in the Advanced information section. The name of a database link must be unique.

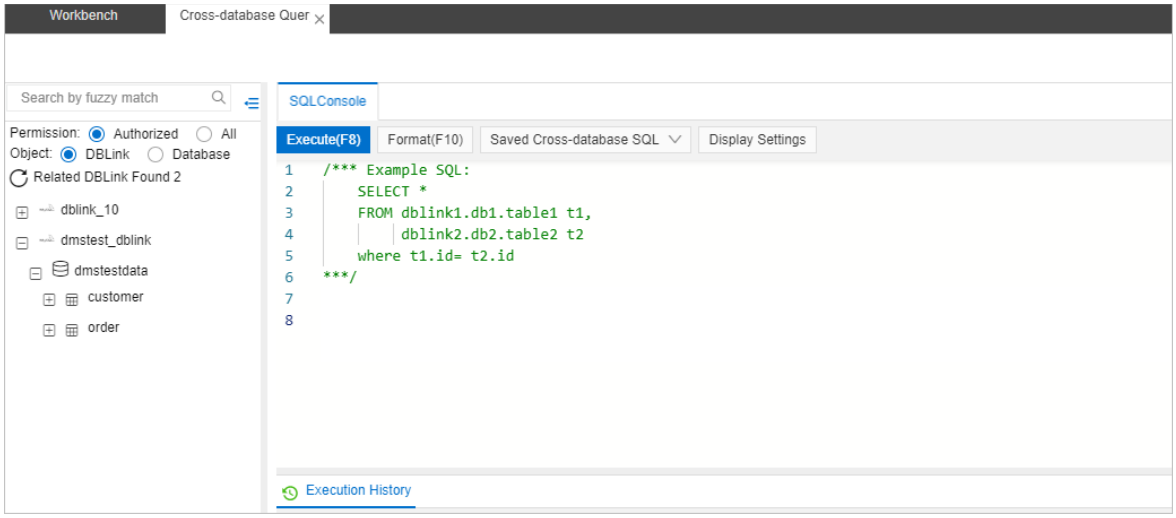
- You have the query permission on each database or table that you want to query.

Limits

You can perform join queries only across physical databases, but not across logical databases.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **SQLConsole > Cross-database Query**.
3. In the left-side pane of the **Cross-database Query** tab, view the databases on which you have the query permission or all the databases.
4. In the right-side SQLConsole, enter an SQL statement. You can perform join queries across databases and tables on which you have the query permission in database instances.



> **Note**
> - You must specify the table that you want to query in the format of `<DBLinkName>.<databaseName>.<tableName>`, such as `dmstest_dblink.dmstestdata.customer`.
> - In the left-side list, you can double-click a table on which you have the query permission or drag the table to the SQLConsole. An SQL statement that you can use to query data in the table is automatically generated.

5. Click **Execute(F8)**. You can view the execution results and execution history in the lower part.

> **Note** DMS also provides the Format, Saved Cross-database SQL, and Display Settings features. You can use these features based on your business requirements.

19.1.11. System management

19.1.11.1. Instance management

DMS allows you to manage database instances. For example, you can export the information about instance configurations or configure a whitelist.

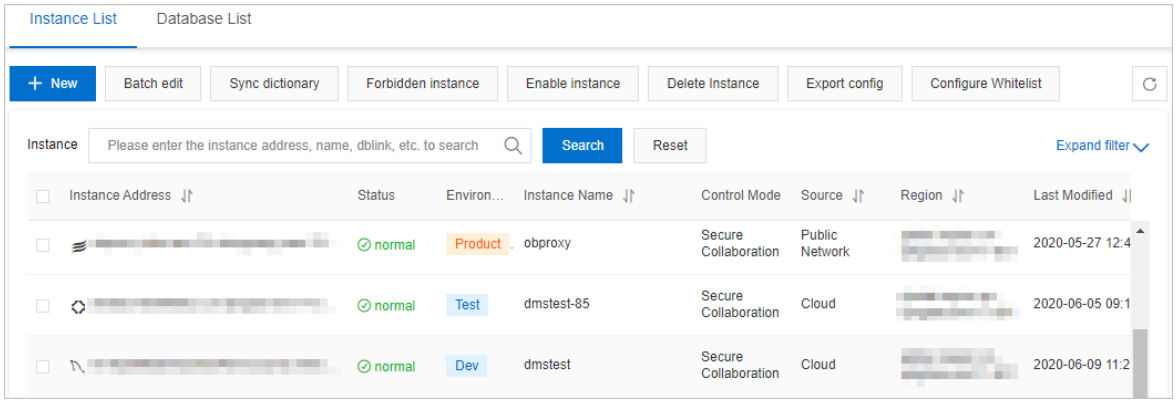
Prerequisites

You are a DBA or a DMS administrator.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **System Management > Instance**.
3. On the **Instance List** tab, select the check box for one or more database instances that you want to manage. Then, you can perform the following operations based on your business requirements:

> **Note** You can click **Expand filter** to show more filter conditions.



- Add an instance

  Click **New** and register a database instance with DMS. For more information, see Register database instances with DMS.

- Edit one or more instances

  Click **Batch edit**. In the dialog box that appears, modify the instance information and click **OK**.

  > **Note** The database instances that you select must be of the same database type, such as MySQL.

- Synchronize the data dictionary

  Click **Sync dictionary**. In the message that appears, click **OK**.

  > **Note**
  > - If you change schemas for a database instance by using DMS, DMS automatically synchronizes the data dictionary of the instance.
  > - If you change schemas for a database instance by using a service other than DMS, you must manually synchronize the data dictionary of the instance.

- Disable or enable one or more instances

  Click **Forbidden instance** or **Enable instance**. In the message that appears, click **OK**.

  > **Note**
  > - After you disable a database instance, the instance is removed from the left-side instance list. DMS users can no longer find databases or tables in this instance in the DMS console.
  > - After you enable a database instance, the instance appears in the left-side instance list. Databases in this instance become available. Relevant permissions that have been granted to DMS users on this instance also become valid.

- Remove one or more instances

  Click **Delete Instance**. In the message that appears, click **OK**. After you remove a database instance, the instance is removed from the left-side instance list. DMS users can no longer use databases in this instance in the DMS console. Relevant permissions that have been granted to DMS users on this instance also become invalid and are revoked.

  > **Note** On the **Instance List** tab, you can find database instances in the Delete state and enable these instances to recover them.

- Export configuration information

  Click **Export config**. The browser automatically downloads a CSV file named *instances*. You can use Excel or a text editor to view this file.

- Configure a whitelist

  Click **Configure Whitelist**. In the message that appears, click **OK**. The IP addresses of DMS are automatically added to the whitelists of the selected database instances.

  > **Note** The database instances that you select must be ApsaraDB instances.

- Other operations

  You can find a database instance and click Details in the Actions column to view the details about databases and tables in this instance. You can also move the pointer over More and perform other operations. For example, you can log on to the instance or edit the instance.

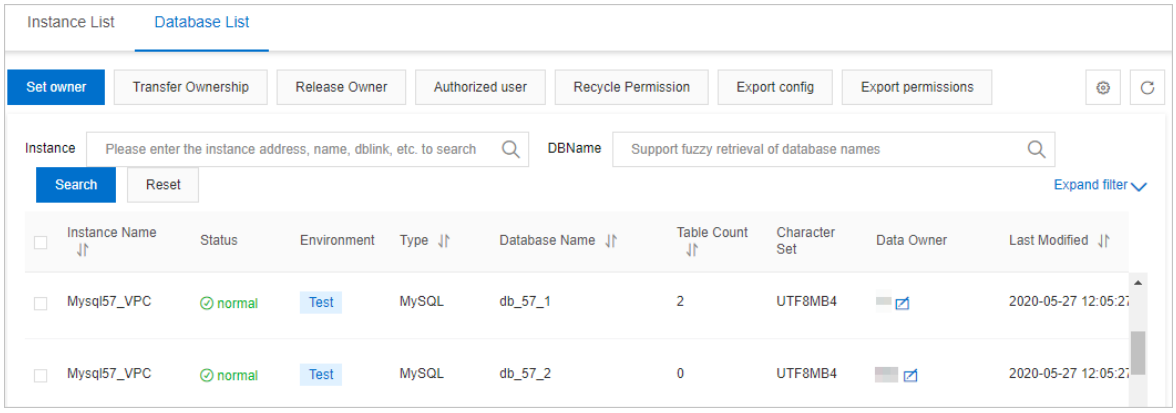
19.1.11.2. Database management

On the Database List tab, you can manage databases. For example, you can set the database owner, transfer the ownership, revoke the owner permission, grant and revoke user permissions, and export the information about database configurations or permissions.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **System Management > Instance**.
3. Click the **Database List** tab.
4. Set conditions to filter databases and select the check box for one or more databases that you want to manage. Then, you can perform the following operations based on your business requirements:

   > **Note** You can click **Expand filter** to show more filter conditions.



   - Set the owner

     Set the owner for the selected databases. You can set multiple owners for multiple databases at a time.

   - Transfer the ownership

     Transfer the ownership of the selected databases to a specific user. If you transfer the ownership of multiple databases at a time, you can select only the common owner of these databases as the original owner.

   - Revoke the owner permission

     Revoke the owner permission from the owners of the selected databases.

- Grant permissions

  Grant the query, export, or change permission on the selected databases to one or more users. In addition, you can set the expiration time for the permissions.

- Revoke permissions

  Revoke the query, export, or change permission on the selected databases from one or more users. If a user does not have the corresponding permissions, the message `No corresponding permissions. You do not need to recycle or release permissions` appears.

- Export configuration information

  Export the configuration information of the selected databases in an Excel file. The configuration information includes the instance status, environment, DBA, and owner.

- Export permission information

  Export the permission information of the selected databases in an Excel file. The permission information includes the database information, users, permissions, and users who grant the permissions.

- Other operations

  You can find a database and click Tables in the Actions column to view the details about tables in this database. You can also move the pointer over More and perform other operations. For example, you can query data in the database, manage permissions, view the details about the database instance to which the database belongs, and find the instance on the Instance List tab.

19.1.11.3. User management

You can use the user management feature to maintain users in DMS. For example, you can adjust the permissions and roles of users.

Prerequisites

You are a DMS administrator.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **System Management > User**.
3. Select the check box for one or more users that you want to manage. Then, you can perform the following operations based on your business requirements:
   - Add a user

     Click **New**. In the dialog box that appears, enter the configuration information of a user. For more information, see Add a user.

   - Edit a user

     Click **Edit User**. In the dialog box that appears, modify the configuration information of the selected user and click **Confirm Change**.

   - Disable or enable one or more users

     Click Operation user and select **Disable User** or **Enable User**. In the message that appears, click **OK**.

   - Remove one or more users

     Click Operation user and select **Delete User**. In the message that appears, click **OK**.

   - Grant permissions

     Click Authorize user and select **Authorize instance**, **Authorize database**, or **Authorize table** in the upper-left corner. In the dialog box that appears, enter a keyword to search for instances, databases, or tables, select the permissions to be granted and the expiration time, and then click **OK**.

- Manage permissions

  Find a user and click **Permission** in the **Actions** column. Set conditions to filter the permissions that are granted to the user. Select the permission that you want to manage and click **Release Permission** to revoke the permission.

19.1.11.4. Task management

The task management feature allows you to manage various tasks that are created by using tickets. You can also directly create or manage tasks.

Prerequisites

You are a DBA or a DMS administrator.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **System Management > Task**.
3. On the **Task** tab, view and manage various tasks that are created by using tickets.
4. Find a task and perform operations based on your business requirements. For example, you can click **Pause**, **Retry**, and **Delete** to perform operations.
   - Pause a task

     Click **Pause** to stop the task.
   - Retry a task

     If the task is in the **Failure** state, click **Retry** to run the task again.
   - Delete a task

     Click **Delete**. The task enters the **Delete** state and is no longer run.
   - Create a task

     Click **Add SQL task**. In the dialog box that appears, enter the task description, database that you want to manage, and SQL statements to be executed. Then, click **Submit Task**.

19.1.11.5. Security management

19.1.11.5.1. Manage security rules

Security rules are defined, by using a domain-specific language (DSL), to control user operations on different types of databases in various aspects, such as syntax and the number of affected rows. You can use security rules to standardize database operations, development processes, and approval processes as needed. This topic describes how to manage security rules.

Prerequisites

You are a DBA or a DMS administrator.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **System Management > Security > Security Rules**.
3. Perform the following operations based on your business requirements:
   - Create a rule set

Click **Create Rule Set**. In the dialog box that appears, set the engine type, enter the name and description of a rule set, and then click **Submit**.

- Edit a rule set
  a. Find a rule set and click **Edit** in the **Actions** column.
  b. Click a rule subset in the left-side pane, such as **SQLConsole**. In the right-side pane, select a checkpoint.
  c. Find a rule, click **Edit**, and then modify the rule. For more information about the syntax, see DSL syntax for security rules.

  > **Note** You can also disable or delete a rule.

- Create a similar rule set
  a. Find a rule set and click **Create As** in the **Actions** column.
  b. In the dialog box that appears, enter the name and description of a new rule set.
  c. Click **Submit**. DMS clones the configurations of the original rule set to the new rule set.

- Delete a rule set

  Find a rule set and click **Delete** in the **Actions** column. In the message that appears, click **OK**.

  > **Note**
  > - A deleted rule set cannot be recovered. We recommend that you proceed with caution.
  > - You can delete only custom rule sets, but not built-in rule sets.

- Set as the default rule set

  Find a rule set and click **Set as Default**. In the message that appears, click **OK**. The rule set is used as the default rule set for the corresponding database engine.

19.1.11.5.2. DSL syntax for security rules

DMS provides a DSL to describe security rules. You can use the DSL syntax to define security rules as needed. This allows you to define database development standards based on your business requirements.

Overview

The DSL syntax is an IF-THEN or IF-THEN-ELSE statement that consists of one or more conditions and actions.

> **Note** When you define a security rule, you must specify an IF condition. You can specify one or more ELSEIF conditions or an ELSE clause as needed.

Example 1: If Condition 1 is met, DMS performs Action 1.

```
if
    Condition 1
then
    Action 1
end
```

Example 2: If Condition 1 is met, DMS performs Action 1. If Condition 2 is met, DMS performs Action 2. If neither Condition 1 nor Condition 2 is met, DMS performs Action 3.

Note If the `else Action 3` clause is omitted, DMS performs no action when neither Condition 1 nor Condition 2 is met.

```
if
    Condition 1
then
    Action 1
elseif
    Condition 2
then
    Action 2
[else Action 3]
end
```

DSL syntax

- Conditional clauses

  DMS uses conditional clauses to evaluate whether to perform actions. The result of a conditional clause is true or false. A conditional clause consists of one or more connectors, operators, and factors. Connectors are AND and OR. Factors are built-in system variables. The following examples are valid conditional clauses:

```
1. true                    // This is the simplest conditional clause. The result is true.
2. 1 > 0
3. 1 > 0 and 2 > 1
4. 1 <= 0 or 1 == 1
```

- Connectors

  Connectors are AND and OR. The AND connector has higher priority than the OR connector. Both the connectors have lower priority than operators. For example, a conditional clause is `1 <= 0 or 1 == 1`. DMS evaluates the result of the `1 <= 0` expression and then the result of the `1 == 1` expression. After that, DMS checks whether at least one of the results of the two expressions is true to evaluate the result of the conditional clause.

- Operators

  Operators are used to connect factors and constants to perform logical operations. The following table describes the operators that are supported by DMS.

Operator	Description	Example
==	Evaluates whether a value is equal to another value.	1 == 1
!=	Evaluates whether a value is not equal to another value.	1 != 2
>	Evaluates whether a value is greater than another value.	1 > 2
>=	Evaluates whether a value is greater than or equal to another value.	1 >= 2
<	Evaluates whether a value is less than another value.	1 < 2
<=	Evaluates whether a value is less than or equal to another value.	1 <= 2

Operator	Description	Example
in	Evaluates whether a value belongs to an array of values.	'a' in ['a', 'b', 'c']
not in	Evaluates whether a value does not belong to an array of values.	'a' not in ['a', 'b', 'c']
matchs	Evaluates whether a string matches a regular expression.	'idxaa' matchs 'idx\w+'
not matchs	Evaluates whether a string does not match a regular expression.	'idxaa' not matchs 'idx\w+'
isBlank	Evaluates whether a value is empty.	'' isBlank
isNotBlank	Evaluates whether a value is not empty.	'' isNotBlank

Note

- If you need to use a backslash (\) in a regular expression, you must use another backslash (\) as an escape character before the backslash. For example, to write the `idx_\w+` expression, you must enter `idx_\\w+`.
- We recommend that you use parentheses () for expressions that you want DMS to evaluate first. For example, a conditional clause is `1 <= 2 == true`. To make the priority clearer, you can change the clause to `(1 <= 2) == true`. DMS first evaluates the result of the `1 <= 2` expression in the parentheses.

- Factors

  A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name is prefixed with `@fac.`. Each tab of the Security Rules tab provides different factors for different checkpoints. The following table describes the factors that are supported by DMS.

Factor	Description
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.
@fac.sql_type	The type of the SQL statement. The value is the subcategory of the SQL statement, such as UPDATE or INSERT. For more information, see the SQL subcategories that are described in the "SQLConsole for relational databases" topic.
@fac.detail_type	The type of the data change. Valid values: ○ COMMON: a Normal Data Modify ticket ○ CHUNK_DML: a Lock-Free Data Modify ticket ○ PROCEDURE: a Programmable Object ticket ○ CRON_CLEAR_DATA: a History Data Clean ticket ○ BIG_FILE: a Large Data Import ticket
@fac.is_logic	A Boolean value that indicates whether the database to be affected is a logical database.

Factor	Description
@fac.extra_info	Other information about the ticket. This factor is not in use.
@fac.is_ignore_affect_rows	A Boolean value that indicates whether to skip the validation.
@fac.insert_rows	The number of rows of data to be inserted.
@fac.update_delete_rows	The number of rows of data to be updated.
@fac.max_alter_table_size	The size of the largest tablespace where the table to be modified is stored.
@fac.is_has_security_column	A Boolean value that indicates whether the SQL statement to be executed involves sensitive fields.
@fac.security_column_list	A list of sensitive fields that the SQL statement to be executed involves.
@fac.risk_level	The risk level of the operation that is to be performed by the SQL statement.
@fac.risk_reason	The reason for identifying the operation to be performed as at the risk level.

Note You can use factors in conditional clauses. For example, you can write `@fac.sql_type == 'DML'` to evaluate whether an SQL statement is a DML statement.

- Action clauses

  An action in a security rule is an operation that DMS performs when the IF condition in the rule is met. For example, DMS can forbid the submission of a ticket, select an approval process, approve a ticket, or reject a ticket. An action in a security rule shows the purpose of the security rule. An action name is prefixed with `@act.` .Each tab of the Security Rules tab provides different actions for different checkpoints. The following table describes the actions that are supported by DMS.

Action	Description
@act.allow_submit	Requires the submission of SQL statements to be executed in a ticket.
@act.allow_execute_direct	Allows the execution of SQL statements in the SQLConsole.
@act.forbid_execute	Forbids the execution of SQL statements.
@act.mark_risk	Marks the risk level of an operation. Example: `@act.mark_risk 'Medium risk level: online environment'`.
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	
@act.choose_approve_template_with_reason	

- Built-in functions

  DMS provides built-in functions that can be used in both conditional clauses and action clauses. A function name is prefixed with `@fun.`.

Function	Description	Example
@fun.concat	Connects strings to form a single string. Output: a string. Input: multiple strings.	@fun.concat('d', 'm', 's') // The output is the string 'dms'. @fun.concat('[Development standards] The [', @fac.column_name, '] field cannot be left empty.') // The output is a message that reminds the user who submits the ticket to enter a value in the field.
@fun.char_length	Calculates the length of a string. Output: an integer. Input: a string.	@fun.char_length('dms') // The output is 3. @fun.char_length(@fac.table_name) // The output is the length of the table name.
@fun.is_char_lower	Evaluates whether all the letters in a string are lowercase. Output: true or false. Input: a string.	@fun.is_char_lower('dms') // The output is true. @fun.is_char_lower(@fac.table_name) // If all the letters in the table name are lowercase, the output is true.
@fun.is_char_upper	Evaluates whether all the letters in a string are uppercase. Output: true or false. Input: a string.	@fun.is_char_upper('dms') // The output is false. @fun.is_char_upper(@fac.table_name) // If all the letters in the table name are uppercase, the output is true.
@fun.array_size	Counts the number of values in an array. Output: an integer. Input: an array of values.	@fun.array_size([1, 2, 3]) // The output is 3. @fun.array_size(@fac.table_index_array) // The output is the number of indexes of the table.
@fun.add	Adds multiple numeric values. Output: a numeric value. Input: multiple numeric values.	@fun.add(1, 2, 3) // The output is 6.
@fun.sub	Deducts a numeric value from another numeric value. Output: a numeric value. Input: two numeric values.	@fun.sub(6, 1) // The output is 5.

Function	Description	Example

@fun.between	Evaluates whether a value belongs to a specific closed range. The supported data types are numeric values, dates, and time. Output: true or false. Input: three values. The first value is the value to be evaluated. The second value indicates the lower limit. The third value indicates the upper limit.	@fun.between(1, 1, 3) // The output is true because the value 1 belongs to [1, 3].
		@fun.between(2, 1, 3) // The output is true because the value 2 belongs to [1, 3].
		@fun.between(7, 1, 3) // The output is false because the value 7 does not belong to [1, 3].
		@fun.between(@fac.export_rows, 2001, 100000) // If the number of exported rows belongs to [2001, 100000], the output is true.
		@fun.between(@fun.current_datetime(), '2019-10-31 00:00:00', '2019-11-04 00:00:00') // If the current date and time belong to [2019-10-31 00:00:00, 2019-11-04 00:00:00], the output is true.
		@fun.between(@fun.current_date(), '2019-10-31', '2019-11-04') // If the current date belongs to [2019-10-31, 2019-11-04], the output is true.
		@fun.between(@fun.current_time(), '13:30:00', '23:59:59') // If the current time belongs to [13:30:00, 23:59:59], the output is true.

@fun.current_datetime	Obtains the current date and time, in the format of yyyy-MM-dd HH:mm:ss. Output: a string. Input: none.	@fun.current_datetime() // For example, the output is 2019-10-31 00:00:00.
@fun.current_date	Obtains the current date, in the format of yyyy-MM-dd. Output: a string. Input: none.	@fun.current_date() // For example, the output is 2020-01-13.
@fun.current_time	Obtains the current time, in the format of HH:mm:ss. Output: a string. Input: none.	@fun.current_time() // For example, the output is 19:43:20.

DSL configuration examples

Limit the number of SQL statements in a ticket: If the number of SQL statements in a ticket exceeds 1,000, DMS rejects the ticket and displays the specified message.

```
if
    @fac.sql_count > 1000
then
    @act.reject_execute 'The number of SQL statements in a ticket cannot exceed 1,000.'
else
    @act.allow_execute
end
```

Allow the submission of only DML statements: If the SQL statements in a ticket are DML statements such as the UPDATE, DELETE, and INSERT statements, DMS allows the execution of these statements.

```
if
    @fac.sql_type in [ 'UPDATE','DELETE','INSERT','INSERT_SELECT']
then
    @act.allow_submit
end
```

19.1.11.5.3. Set security rules for an instance

This topic describes how to set security rules for a database instance.

Prerequisites

- You are a DBA or a DMS administrator.
- The control mode of a database instance is **Security Collaboration**.

Procedure

1. Log on to the DMS console.
2. In the left-side navigation pane, right-click the database instance that you want to manage.
3. In the shortcut menu that appears, choose **Control Mode > Security Collaboration** and select the security rule set that you want to use.

> **Note** You can also edit the database instance on the Instance tab and change the security rule set for the instance. For more information, see Instance management.

19.1.11.5.4. Customize approval processes

DMS allows you to configure instance-level security rules so that you can customize different approval processes for different database instances or database operations. However, instance-level security rules may have specific limits in the production environment. This topic describes how to customize an approval process.

Prerequisites

You are a DBA or a DMS administrator.

Context

- Each database instance has only one DBA, who is responsible for approving tickets. The entire approval process may get stuck when the DBA is not able to respond as expected. To avoid this issue, you may need to allow multiple DBA roles to participate in an approval process.
- If multiple business parties share the same database in a database instance, each business party may need to approve tickets for their respective business operations in an approval process.

To resolve the issues in the preceding scenarios, you can customize approval processes.

Usage notes

- Do not assign only one approver to an approval node. We recommend that you assign at least two approvers to each approval node and assign at least two data owners for a database.
- You can assign a maximum of three data owners for a database. If multiple business parties share the same database, you can allow all business parties to participate in an approval process by following the procedure in this topic: Create an approval node and add the data owners of multiple business parties as approvers. Then, add the new node instead of the system node Owner to an approval template.

Procedure

This topic describes the procedure for customizing an approval process by allowing multiple DBA roles to participate in an approval process. You can follow this procedure to customize an approval process in other scenarios.

1. Log on to the DMS console.
2. In the top navigation bar, choose **System Management > Security > Approval Processes**.
3. Create an approval node.

i. Click the **Approval Node** tab on the left. Then, click **Create Approval Node**.

ii. Configure the approval node.

Parameter	Description
Node Name	The name of the approval node. The name must be globally unique.
Remarks	The description of the approval node, which helps distinguish the node from the others.
Approver	The approvers on the current node. Select the Apsara Stack tenant accounts or RAM users of relevant approvers. You can enter a keyword to select an account from the auto-completion list. **Note** In this example, three approvers are selected.

iii. Click **Submit**.

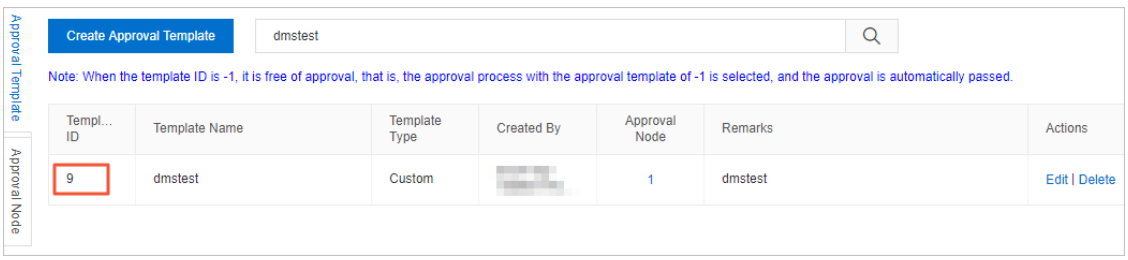
4. Create an approval template.

i. Click the **Approval Template** tab on the left. Then, click **Create Approval Template**.

ii. Configure the approval template.

Parameter	Description
Template Name	The name of the approval template. The name must be globally unique.
Remarks	The description of the approval template, which helps distinguish the template from the others.
Approval Node	Click Add Node and select the required approval nodes. In this example, the system node Owner and the approval node that is created in Step 3 are selected so that multiple DBA roles can participate in the approval process. **Note** The approval process is implemented based on the value of the Approval Order parameter in ascending order.

iii. Click **Submit**.
After the approval template is created, you can view the ID of the approval template. In this example, the ID is 9.



5. Apply the new approval process.

This example shows you how to edit the rule of the medium-level risk approval process for the **Risk Approval Rules** checkpoint in the data change approval process. You can follow this procedure to apply an approval process in other scenarios.

i. In the top navigation bar, choose **System Management > Security > Security Rules**.

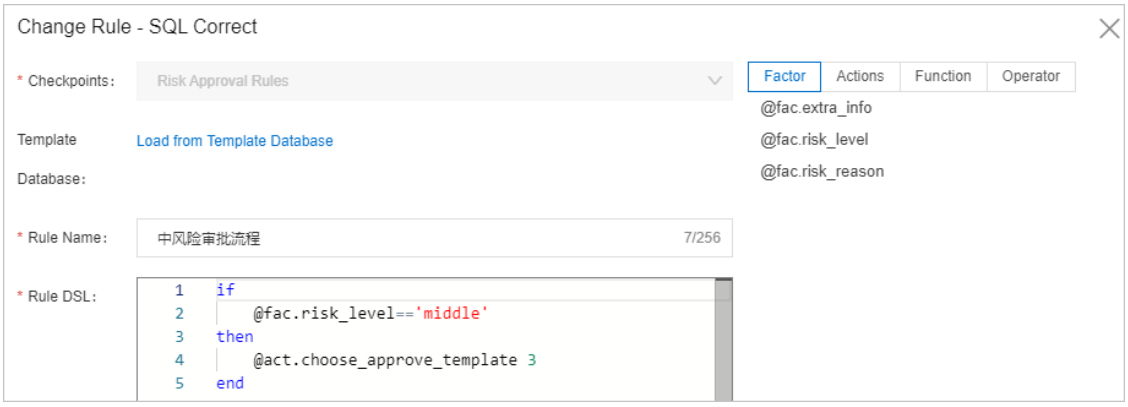
ii. Find the rule set that you want to edit and click **Edit** in the **Actions** column.

iii. In the left-side pane, click the **SQL Correct** tab.

iv. Select **Risk Approval Rules** as the checkpoint.

v. Find the rule of the medium-level risk approval process and click **Edit**.

vi. In the **Rule DSL** field, change the template ID.



> **Note** In this example, change `3` in the preceding figure to `9`, which is the ID of the approval template that is created in Step 4.

vii. Click **Submit**.

Result

If the data change tickets that are subsequently submitted match the corresponding rule, all of the specified DBA roles will receive ticket approval notifications and can participate in the approval process.

19.1.11.5.5. View operations logs

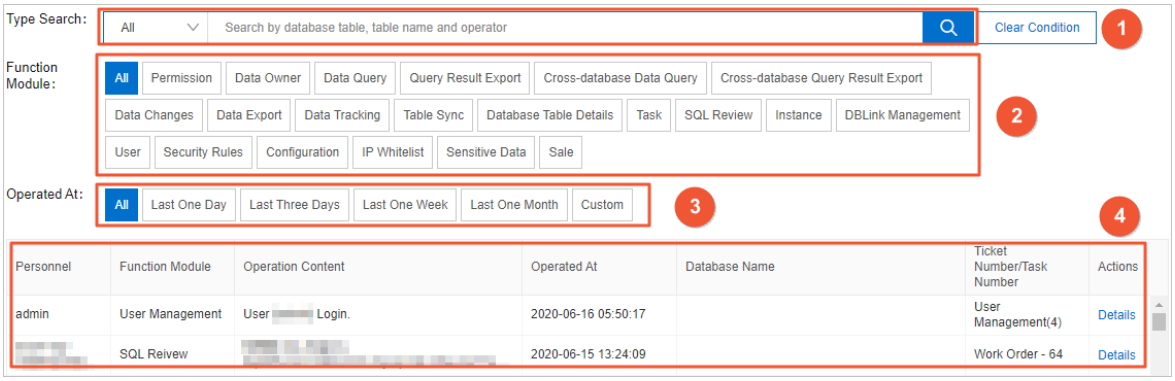
Operations logs in DMS record data changes. Each record contains information such as the user, specific operation, and time at which the operation was complete. You can trace historical user operations at any time.

Prerequisites

You are a security administrator, a DBA, or a DMS administrator.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **System Management > Security > Operation Logs**.
3. On the **Operation Logs** tab, set conditions to filter logs and view the logs.



No.	Section	Description
①	**Type Search**	Allows you to search for logs by keyword, such as the database name, table name, or operator.
②	**Function Module**	Allows you to search for logs by feature.
③	**Operated At**	Allows you to search for logs by time period based on the time when an operation was complete.
④	Log list	Displays the logs that are found based on the specified filter conditions. By default, the logs are listed in reverse order of the recorded time. You can find a log and click **Details** in the **Actions** column to view the details about the log.

19.1.11.5.6. Configure access IP address whitelists

DMS allows you to configure access IP address whitelists to effectively control the service scope of DMS. You can allow user access to DMS only from specific trusted network environments.

Prerequisites

You are a DMS administrator.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **System Management > Security > Access IP Whitelists**.
3. Perform the following operations based on your business requirements:
   - Enable or disable the whitelist control feature

     Click **Click to Open** or **Click to Close** to enable or disable the whitelist control feature.
   - Create a whitelist
     a. Click **Create Whitelist**.
     b. In the dialog box that appears, enter the IP addresses and description.

     > **Note**
     > - Separate IP addresses with semicolons (;). Ensure that each IP address is unique in a whitelist.
     > - Supported formats are IP addresses such as 10.23.12.24, or CIDR blocks such as 10.23.12.24/24. /24 indicates the length of the IP address prefix in the CIDR block. An IP address prefix can contain 1 to 32 bits.
     > - 0.0.0.0/0 indicates that all IP addresses are allowed.

     c. Click **Submit**.
   - Edit a whitelist
     a. Find an IP address whitelist and click **Edit** in the **Actions** column.
     b. In the dialog box that appears, modify the IP address information.
     c. Click **Submit**.
   - Delete a whitelist
     a. Find an IP address whitelist and click **Delete** in the **Actions** column.

b. In the message that appears, click **OK**.

> **Note** You cannot delete all IP address whitelists. At least one IP address whitelist must be retained.

19.1.11.5.7. Configure row-level control

In specific scenarios, different users have access to different rows of data in the same table. Traditional solutions address this need by using views. DMS provides the row-level control feature to control permissions at the row level.

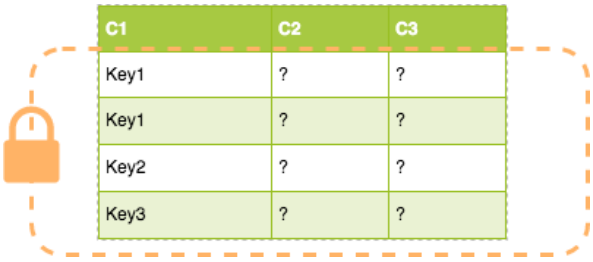
Prerequisites

You are a security administrator, a DBA, or a DMS administrator.

Context

Row-level control is used to provide horizontal data protection for tables. All the rows of a table are distinguished by one or more specified values. These values are called control values. To have access to a row that corresponds to a control value in the DMS console, users must have permissions on the row.

> **Note** A control value may correspond to multiple rows. If a user has permissions on a control value that corresponds to multiple rows, the user has permissions on all the rows that correspond to this control value.



Limits

- The sensitive data management feature applies only to relational databases such as MySQL, but not to NoSQL databases.
- The database instance to which you want to apply this feature must be managed in Security Collaboration mode.
- This feature applies only to physical databases, but not to logical databases.
- If you want to query, modify, or delete data in row-level control tables, the following limits are imposed on SQL statements:
  i. The control field must be specified in SQL statements.
  ii. All the rows of a row-level control table require access permissions. Users who do not have permissions on all rows can use only the `=` and `IN` operators to specify the control field. The control value that is specified in an SQL statement must belong to the set of control values of the table.
  iii. Users who do not have permissions on all rows cannot use specific operators such as OR, XOR, and logical NOT.

Terms

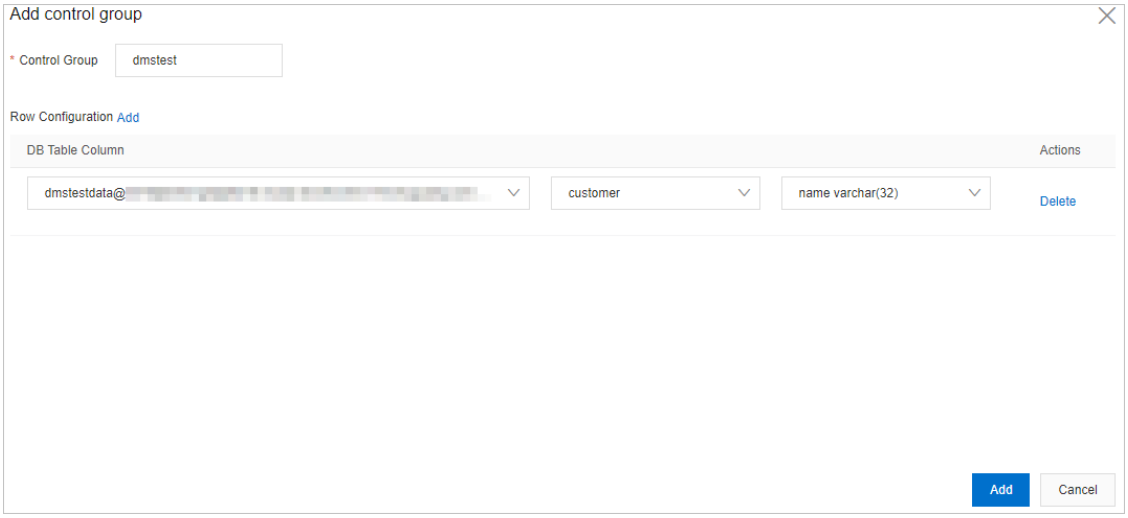
Term	Description

Term	Description
row permission	Users can apply for permissions on a control value to have access to rows that correspond to the control value. Permissions on the rows of a table are defined as row permissions and incorporated into the existing permissions of DMS. Permissions that can be controlled in Security Collaboration mode include database, table, column or field, and row permissions.
Single control value	When a user applies for permissions on the rows of a row-level control table, the user can select **Single** to apply for permissions on a single control value. **Note** A control value may correspond to multiple rows. If a user has permissions on a control value that corresponds to multiple rows, the user has permissions on all the rows that correspond to this control value.
All control values	When a user applies for permissions on the rows of a row-level control table, the user can select **ALL** to apply for permissions on all control values. Then, the user can have permissions on all the rows of the table. In addition, the user can have access to the entire row-level control table without limits. Even if the control values are changed or more control values are added, the user still has permissions on all the rows of the table.
row-level control table	A table that requires row-level control is called a row-level control table.
control field	The control values of a row-level control table are added to a field and this field is called a control field.
control group	If multiple row-level control tables have the same control values, these tables can be put in a control group. For example, if Table A and Table B have the same control values, the two tables can be put in a control group and managed at the same time by using only one set of control values.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **System Management > Security > Sensitive Data**.
3. Click the **Row Level Security** tab on the left.
4. Create a control group.
    i. Click **Add control group**.

ii. In the dialog box that appears, configure the control group.



Parameter	Description
Control Group	Enter the name of the control group.
Row Configuration	Click **Add** to add a row configuration where you can specify the database, table, and field. **Note** You can repeat this operation to add multiple row configurations.
DB Table Column	Enter the keyword of a database name in the database field to search for databases and select a database. Then, select a table and a field from the respective drop-down lists. **Note** The selected field is the control field.

iii. Click **Add**.

5. Add control values.
   i. Find the control group that is created and click **Details** in the **Actions** column.
   ii. Click **Add Row Value**.
   iii. In the dialog box that appears, specify whether to append row values and enter the row values.

   Note Separate multiple row values with commas (,).

   iv. Click **Import**.

Related operations

After you configure row-level control for a table, if a user does not have permissions on a control value that corresponds to one or more rows in the table, the user has no permission to query data in the rows. The user can apply for permissions on the control value to obtain access to the rows. For more information, see Apply for permissions.

19.1.11.5.8. Manage sensitive data

DMS allows you to manage all sensitive and confidential fields in a uniform manner. You can configure redaction algorithms for sensitive and confidential fields. This enhances the data de-identification feature.

Prerequisites

You are a security administrator, a DBA, or a DMS administrator.

Context

When you query a table and the table contains sensitive or confidential fields on which you do not have permissions, the values of the fields are displayed as `******` in query results. In this case, sensitive data is fully redacted. In specific scenarios, developers or testers may need to view a part of sensitive data for troubleshooting. To meet this requirement, you can configure redaction algorithms for fields.

Limits

- The sensitive data management feature applies only to relational databases such as MySQL, but not to NoSQL databases.
- The database instance to which you want to apply this feature must be managed in Security Collaboration mode.

Procedure

1. Log on to the DMS console.
2. Set the security levels of fields in the table that you want to manage.

   > **Note** If you have set the security levels of fields, you can skip this step.

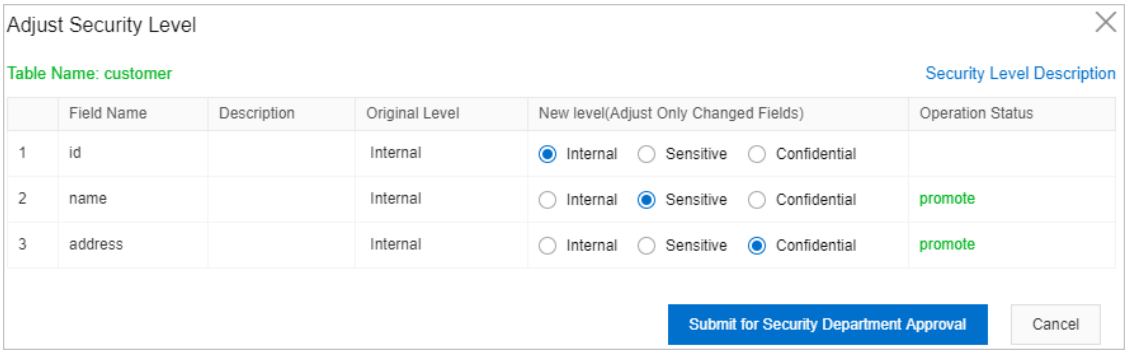
   i. In the left-side database instance list, find the database instance that you want to manage and click the icon to show the list of databases in this instance.

   ii. Find the database that you want to manage, right-click the database, and then select **Tables**.

   iii. Find the table that you want to manage and click the + icon before the table name to show the table details.

   iv. Click **Adjust**.

   v. In the dialog box that appears, adjust the security levels of fields.

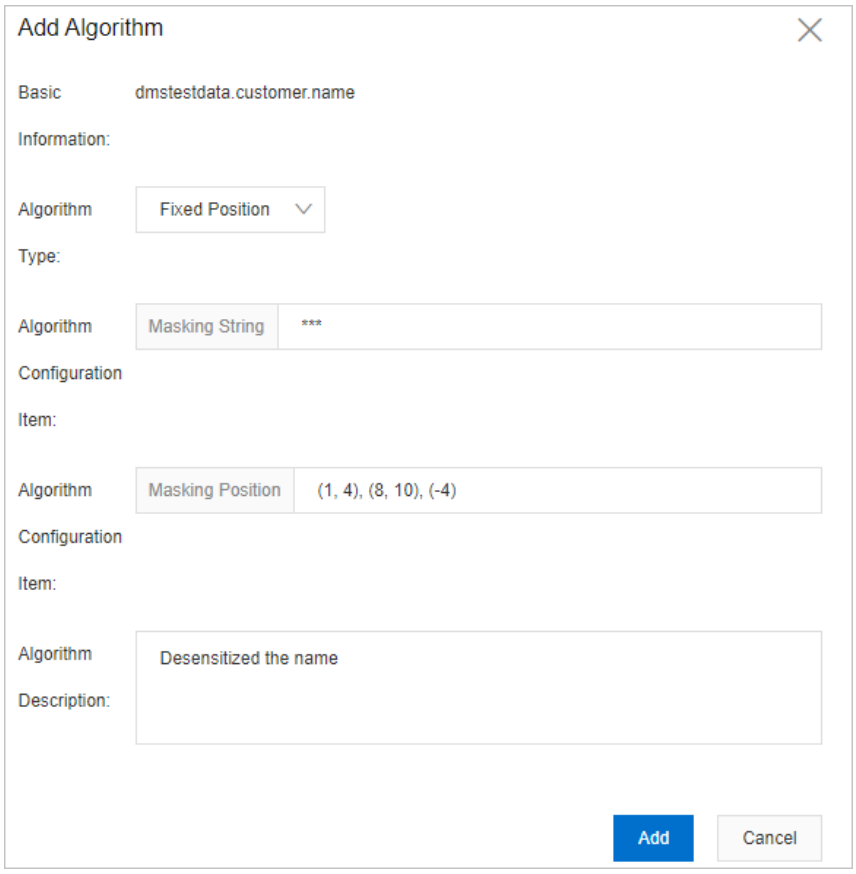
   

   vi. Click **Submit for Security Department Approval**.

   > **Note** DMS automatically approves applications for raising the security level of a field. Applications for lowering the security level of a field must be approved based on the approval process that is specified by the DMS administrator or DBA.

   vii. In the message that appears, click **OK**.

3. In the top navigation bar, choose **System Management > Security > Sensitive Data**.
4. Find a field and click **Add Algorithm** in the **Actions** column.
5. In the dialog box that appears, configure a redaction algorithm.



Parameter	Description
Algorithm Type	Select an algorithm type based on your business requirements.

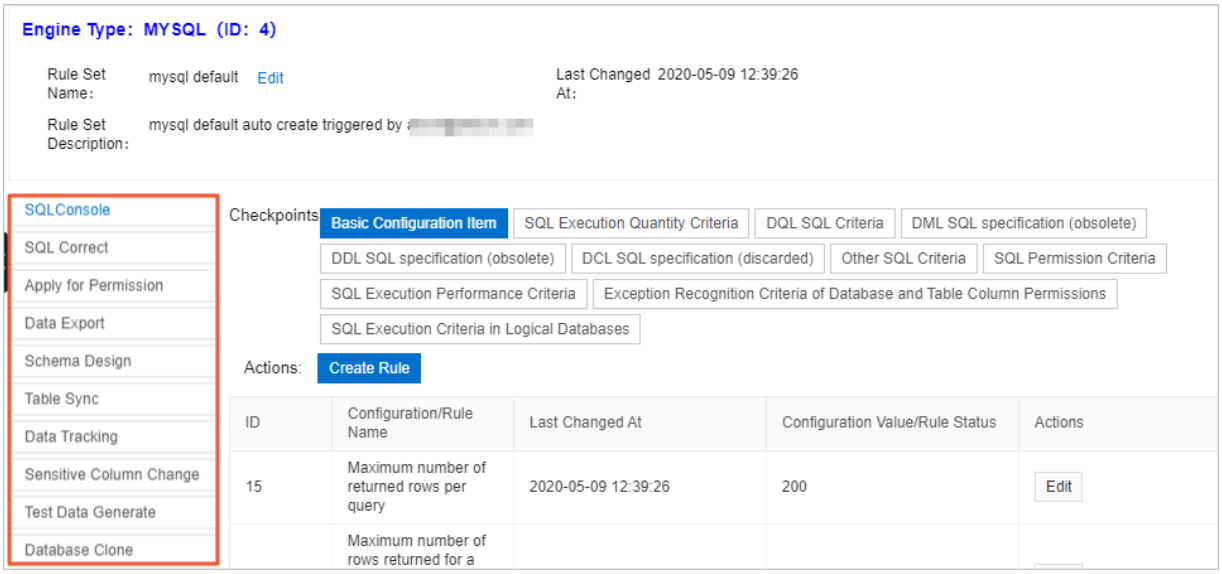
Parameter	Description
Algorithm Configuration Item	The algorithm configuration items vary based on the algorithm type that you select: ○ Select **Fixed Position** as the algorithm type You must set the Masking String and Masking Position parameters. For example, you can set the Masking String parameter to ***. The Masking Position parameter specifies the positions of the characters to be redacted in the field values. The positions are in the format of coordinates. Examples: ■ (1, 4): redacts the first four characters, which are the first to fourth characters. You can also enter (4) for short. ■ (-4): redacts the last four characters. **Note** You can specify a maximum of three positions. For example, `(1, 4), (8, 10), (-4)` indicates to redact the first four characters, the eighth to tenth characters, and the last four characters. ○ Select **Fixed Character** as the algorithm type You must set the Masking String and Character to Be Replaced parameters. **Note** The Character to Be Replaced parameter specifies the characters to be redacted, in the format of a string. You can specify a maximum of three strings. ○ Select **Full Masking** as the algorithm type You need to set only the Masking String parameter.
Algorithm Description	Enter an informative description, which helps distinguish the algorithm from the others.

6. Click **Add**.

19.1.11.6. Security rules

19.1.11.6.1. Overview of security rule sets

Security rule sets are sets of rules that use a DSL to achieve the fine-grained management of databases. These rules apply to different database types, different database management statements, and different scopes of data changes in databases. By configuring these rules, you can formulate operation guidelines and define development and approval processes based on your business requirements.



The following section lists the topics that describe the security rules that are provided on different tabs of the Security Rules tab. You can click a link to view the information about the security rules on the corresponding tab, including the basic configuration items, checkpoints, factors, actions, and supported statements or commands.

- SQLConsole for relational databases
- SQLConsole for MongoDB
- SQLConsole for Redis
- Data change
- Permission application
- Data export
- Schema design
- Database and table synchronization
- Schema design
- Sensitive field change
- Test data generation
- Database cloning

19.1.11.6.2. Manage the security rules under checkpoints

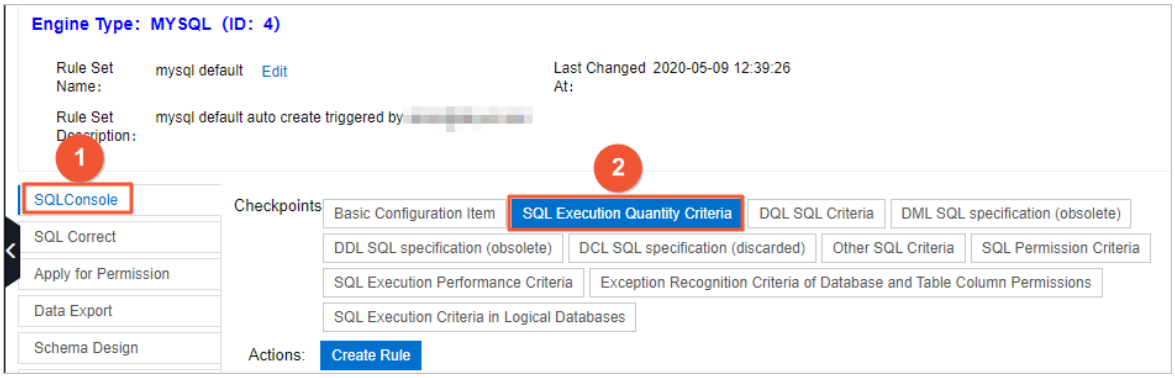
This topic describes how to configure a security rule under the SQL Execution Quantity Criteria checkpoint on the SQLConsole tab. The methods to configure the security rules under other checkpoints are similar to the method that is described in this topic.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **System Management > Security > Security Rules**.
3. Find the security rule set that you want to manage and click **Edit** in the **Actions** column.

   > **Note** In this example, the security rule set for MySQL databases is used.

4. Click a tab and then a checkpoint based on your business requirements. In this example, click the **SQLConsole** tab and then the **SQL Execution Quantity Criteria** checkpoint.



> **Note**
> - For more information about the tabs and the corresponding checkpoints, see Overview of security rule sets.
> - You can click **Create Rule** to create a security rule. For more information about the supported syntax, see DSL syntax for security rules.

5. Find the security rule that you want to manage and click **Edit** in the **Actions** column.

> **Note** You can also click **Disable** to disable a security rule or **Delete** to delete a security rule.

6. In the Change Rule - SQLConsole dialog box, modify the DSL statements of the security rule. For more information, see DSL syntax for security rules. In this example, adjust the maximum number of SQL statements that can be executed at a time from 1,000 to 500.

> **Note**
> - Various security rule templates are provided for each checkpoint. You can click **Load from Template Database** to use a template.
> - For more information about factors and actions, see Overview of security rule sets.

7. Click **Submit**.

19.1.11.6.3. SQLConsole for relational databases

DMS allows you to manage relational and non-relational databases on the SQLConsole tab. The definition and classification of security rules are different for relational and non-relational databases. This topic describes the security rules for relational databases on the SQLConsole tab, such as MySQL databases.

Default security rules

- Constraints on SQL statement categories: No constraints are imposed on data query language (DQL) statements. By default, DML statements, DDL statements, data control language (DCL) statements, and SQL statements that cannot be identified by DMS are all blocked. To execute DML, DDL, or DCL statements on the SQLConsole tab, you must configure and enable corresponding security rules.
- Constraints on permissions on databases, tables, and fields: By default, users can perform operations on databases, tables, and fields without permission validation. To enable permission validation, you must configure and enable security rules under the **SQL Permission Criteria** checkpoint. For more information, see Supported checkpoints.

Basic configuration items

Configuration item	Description
Maximum number of returned rows per query	The maximum number of rows that can be returned for a query.
Maximum number of rows returned for a single query with sensitive column conditions	The maximum number of rows that can be returned for a query that contains query conditions for sensitive fields.
Limit the maximum allowed SQL full table scan (MB)	The maximum size of data that can be scanned. Before an SQL statement is executed, DMS checks the execution plan. If the size of the data to be scanned exceeds the specified threshold, the SQL statement fails to be executed. **Note** This item can be configured only for MySQL and Oracle databases.
Turn off the execution of change SQL validation affects the number of rows and prompts	Specifies whether to check the number of rows to be affected and display a prompt before DMS executes an SQL statement to change data. By default, this item is disabled.
How many rows does result set page support	The maximum number of rows that can be returned in the query result set on the SQLConsole tab.
Does the result set support paging	Specifies whether the query result set can be displayed on multiple pages on the SQLConsole tab.
Does the result set support editing	Specifies whether the query result set can be edited on the SQLConsole tab.

Supported checkpoints

Note Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description
SQL Execution Quantity Criteria	Allows you to limit the number of SQL statements that can be submitted at a time.
DQL SQL Criteria	Allows you to set constraints on DQL statements.
Other SQL Criteria	Allows you to set constraints on multiple categories of SQL statements. Different enterprises may define different high-risk SQL statements, which may include specific subcategories of DML, DCL, and DDL statements. **Note** You can also set constraints on SQL statements that cannot be identified by DMS.
SQL Permission Criteria	Allows you to set constraints on the execution of SQL statements from the aspect of permissions. For example, DMS checks whether a user has the required permissions on the corresponding databases, tables, and fields.

Checkpoint	Description
SQL Execution Performance Criteria	Allows you to set constraints on the execution of SQL statements from the aspect of performance. For example, you can specify that a DML statement is not executed if the number of rows to be affected by the statement exceeds the specified threshold, or that a DDL statement is not executed if the size of the table involved exceeds the specified threshold.
Exception Recognition Criteria of Database and Table Column Permissions	After a user submits SQL statements on the SQLConsole tab, DMS parses the SQL statements and checks whether the user has the required permissions on the corresponding databases, tables, and fields. You can configure security rules under this checkpoint to ensure that if exceptions occur when DMS parses complex SQL statements, these statements can be executed. **Note** If you configure and enable security rules under the **Exception Recognition Criteria of Database and Table Column Permissions** checkpoint, security rules under the SQL Permission Criteria, DQL SQL Criteria, Other SQL Criteria, and SQL Execution Performance Criteria checkpoints are automatically disabled.
SQL Execution Criteria in Logical Databases	This checkpoint is reserved for logical databases and not suitable for physical databases.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`.The following table describes the supported factors for relational databases on the SQLConsole tab.

Factor	Description
@fac.sql_count	The number of SQL statements that are submitted at a time.
@fac.select_sql_count	The number of DQL statements among the SQL statements that are submitted at a time.
@fac.dml_sql_count	The number of DML statements among the SQL statements that are submitted at a time.
@fac.sql_type	The category and subcategory of the SQL statement. For more information, see Supported SQL statements.
@fac.sql_sub_type	
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.
@fac.fulltable_delete	A Boolean value that indicates whether the current SQL statement deletes a full table. Valid values: *true* and *false*.
@fac.fulltable_update	A Boolean value that indicates whether the current SQL statement updates a full table. Valid values: *true* and *false*.
@fac.current_sql	The current SQL statement.
@fac.user_is_admin	A Boolean value that indicates whether the current user is a DMS administrator. Valid values: *true* and *false*.
@fac.user_is_dba	A Boolean value that indicates whether the current user is a DBA. Valid values: *true* and *false*.

Factor	Description
@fac.user_is_inst_dba	A Boolean value that indicates whether the current user is the DBA of the current instance. Valid values: *true* and *false*.
@fac.user_is_sec_admin	A Boolean value that indicates whether the current user is a security administrator. Valid values: *true* and *false*.
@fac.sql_affected_rows	The number of rows to be affected by the current SQL statement. **Warning** This factor triggers COUNT operations, which may affect the database performance. Use this factor with caution.
@fac.sql_relate_table_store_size	The estimated total size of the table to be accessed by the current SQL statement. Unit: MB. **Note** This value is estimated based on the metadata that is obtained by DMS. It is not an actual value.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions for relational databases on the SQLConsole tab.

Action	Description
@act.reject_execute	Rejects the request to execute the current SQL statement.
@act.allow_execute	Allows the current SQL statement to be executed.
@act.reject_sql_type_execute	Rejects the request to execute a specific subcategory of SQL statements. You must specify an SQL statement subcategory after the action name. Example: `@act.reject_sql_type_execute 'UPDATE'`.
@act.allow_sql_type_execute	Allows a specific subcategory of SQL statements to be executed. You must specify an SQL statement subcategory after the action name. Example: `@act.allow_sql_type_execute 'UPDATE'`.
@act.check_dml_sec_column_permission	Checks whether a user has the required permissions on sensitive fields. If the user does not have the permissions, the DML statement for data change is not executed.
@act.uncheck_dml_sec_column_permission	Does not check whether a user has the required permissions on sensitive fields.
@act.check_sql_access_permission	Checks whether a user has the required permissions, such as query and change permissions, on the databases, tables, and fields that are involved in the SQL statements to be executed.
@act.uncheck_sql_access_permission	Does not check whether a user has the required permissions on the objects that are involved in the SQL statements to be executed.

Action	Description
@act.enable_sec_column_mask	De-identifies sensitive fields in query result sets that are returned for SQL statements that are submitted by users who do not have permissions on the sensitive fields.
@act.disable_sec_column_mask	Does not de-identify sensitive fields in query result sets that are returned for SQL statements that are submitted by users who do not have permissions on the sensitive fields.

Supported SQL statements

Category	Subcategory
DQL	• SELECT • DESC • EXPLAIN • SHOW
DML	• INSERT • INSERT_SELECT • REPLACE • REPLACE_INTO • UPDATE • DELETE • MERGE

Category	Subcategory
DDL	• DATABASE_OP • CREATE • CREATE_INDEX • CREATE_VIEW • CREATE_SEQUENCE • CREATE_TABLE • CREATE_SELECT • TRUNCATE • DROP_INDEX • DROP_VIEW • DROP_TABLE • RENAME • ALTER • ALTER_INDEX • ALTER_VIEW • ALTER_TABLE • ALTER_SEQUENCE • CREATE_FUNCTION • CREATE_PROCEDURE • ALTER_FUNCTION • ALTER_PROCEDURE • DROP_FUNCTION • DROP_PROCEDURE
DCL	• GRANT • DECLARE • SET • ANALYZE • FLUSH • OPTIMIZE • KILL

19.1.11.6.4. SQLConsole for MongoDB

DMS allows you to manage relational and non-relational databases on the SQLConsole tab. The definition and classification of security rules are different for relational and non-relational databases. This topic describes the security rules for MongoDB databases on the SQLConsole tab.

Basic configuration items

Maximum number of returned rows per query: the maximum number of rows that can be returned for a query.

Supported checkpoints

Note Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description
User Permission Validation	Allows you to specify whether to check the permissions of specific users when they submit commands.
Collection Statement Criteria	Allows you to specify whether to allow DMS to run a specific category of commands.
DB Statement Criteria	
Cache Query Statement Criteria	
User Management Statement Criteria	
Role Management Statement Criteria	
Replication Set Statement Criteria	
Sharding Statement Criteria	

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as command categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`.The following table describes the supported factors for MongoDB databases on the SQLConsole tab.

Factor	Description
@fac.sql_sub_type	The subcategory of the current command. For more information about the supported commands, see Supported MongoDB commands.
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as `DEV` or `PRODUCT`.
@fac.current_sql	The current command.
@fac.user_is_admin	A Boolean value that indicates whether the current user is a DMS administrator. Valid values: *true* and *false*.
@fac.user_is_dba	A Boolean value that indicates whether the current user is a DBA. Valid values: *true* and *false*.
@fac.user_is_inst_dba	A Boolean value that indicates whether the current user is the DBA of the current instance. Valid values: *true* and *false*.
@fac.user_is_sec_admin	A Boolean value that indicates whether the current user is a security administrator. Valid values: *true* and *false*.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions for MongoDB databases on the SQLConsole tab.

Action	Description
@act.reject_execute	Rejects the request to run the current command.
@act.allow_execute	Allows the current command to be run.
@act.reject_sql_type_execute	Rejects the request to run a specific subcategory of commands. You must specify a subcategory after the action name. Example: `@act.reject_sql_type_execute 'UPDATE'`.
@act.allow_sql_type_execute	Allows a specific subcategory of commands to be run. You must specify a subcategory after the action name.

Supported MongoDB commands

Category	Subcategory	Command
Collection commands	Query commands	• aggregate • find • findOne • count • distinct • getIndexes • getShardDistribution • isCapped • stats • dataSize • storageSize • totalIndexSize • totalSize
	Data update commands	• insert • save • findAndModify • remove • update
	Collection modification commands	• drop • renameCollection
	Index modification commands	• createIndex • createIndexes • dropIndexes • reIndex
	Other commands	validate

Category	Subcategory	Command
Database commands	Database query commands	• commandHelp • currentOp • getCollectionInfos • getCollectionNames • getLastError • getLastErrorObj • getLogComponents • getPrevError • getProfilingStatus • getReplicationInfo • getSiblingDB • help • isMaster • listCommands • printCollectionStats • printReplicationInfo • version • serverBuildInfo • serverStatus,stats
	Collection creation commands	createCollection
	High-risk commands	• dropDatabase • fsyncLock • fsyncUnlock • killOp • repairDatabase • resetError • runCommand
Commands related to the query plan cache	Read commands	• getPlanCache • getPlansByQuery • listQueryShapes
	Write commands	clearPlansByQuery
User management commands	User query commands	• getUser • getUsers

Category	Subcategory	Command
	User modification commands	• createUser • changeUserPassword • dropUser • dropAllUsers • grantRolesToUser • revokeRolesFromUser • updateUser
Role management commands	Role query commands	• getRole • getRoles
	Role modification commands	• createRole • dropRole • dropAllRoles • grantPrivilegesToRole • revokePrivilegesFromRole • revokeRolesFromRole • updateRole
Replica set commands	N/A	• help • printReplicationInfo • status • conf
Sharding commands	N/A	• getBalancerState • isBalancerRunning

19.1.11.6.5. SQLConsole for Redis

DMS allows you to manage relational and non-relational databases on the SQLConsole tab. The definition and classification of security rules are different for relational and non-relational databases. This topic describes the security rules for Redis databases on the SQLConsole tab.

Supported checkpoints

Note Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description
Permission Execution Statement Criteria	Allows you to set constraints on the permissions for command execution.
Statement Criteria: Keys	Allows you to specify whether to check the permissions of specific users when they submit commands.

Checkpoint	Description
Statement Criteria: String	Allows you to specify whether to allow the execution of various Redis commands.
Statement Criteria: List	
Statement Criteria: SET	
Statement Criteria: SortedSet	
Statement Criteria: Hash	
Statement Criteria: Other	

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`.The following table describes the supported factors for Redis databases on the SQLConsole tab.

Factor	Description
@fac.cmd_type	The type of the Redis command. For more information about valid values, see Supported Redis commands.
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as `DEV` or `PRODUCT`.
@fac.is_read	A Boolean value that indicates whether the current command is a read command. Valid values: *true* and *false*.
@fac.is_write	A Boolean value that indicates whether the current command is a write command. Valid values: *true* and *false*.
@fac.current_sql	The current command.
@fac.user_is_admin	A Boolean value that indicates whether the current user is a DMS administrator. Valid values: *true* and *false*.
@fac.user_is_dba	A Boolean value that indicates whether the current user is a DBA. Valid values: *true* and *false*.
@fac.user_is_inst_dba	A Boolean value that indicates whether the current user is the DBA of the current instance. Valid values: *true* and *false*.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions for Redis databases on the SQLConsole tab.

Action	Description
@act.reject_execute	Rejects the request to run the current command.
@act.allow_execute	Allows the current command to be run.

Supported Redis commands

Category	Subcategory	Command
Key-related commands	Key-related read commands	• EXISTS • TTL • PTTL • RANDOMKEY • TYPE • SCAN • OBJECTS
	Key-related write commands	• DEL • DUMP • EXPIRE • EXPIREART • MOVE • PERSIST • PEXPIRE • PEXPIREAT • RENAME • RENAMENX • RESTORE • SORT • TOUCH • UNLIMK • WAIT • MIGRATE
	String-related read commands	• GET • GETRANGE • BITCOUNT • GETBIT • MGET • STRLEN • BITOPS

Category	Subcategory	Command
String-related commands	String-related write commands	• APPEND • BITFIELD • BITOP • DECR • DECRBY • GETSET • INCR • INCRBY • INCRBYFLOAT • MSET • MSETNX • PSETEX • SET • SETNX
List-related commands	List-related read commands	• LINDEX • LLEN • LRANGE
	List-related write commands	• BLPOP • BRPOP • BRPOPLPUSH • LINSERT • LPOP • LPUSH • LPUSHX • LREM • LSET • LTRIM • RTOP • RPOPLPUSH • RPUSH • RPUSHX
Set-related commands	Set-related read commands	• SCARD • SISMEMBER • SRANDMEMBER • SSCAN
	Set-related write commands	• SADD • SMOVE • SPOP • SREM

Category	Subcategory	Command
Sorted set-related commands	Sorted set-related read commands	• ZCARD • ZCOUNT • ZLEXCOUNT • ZRANGE • ZRANGEBYLEX • ZRANGEBYSCORE • ZRANK • ZREVRNGE • ZREVRANGEBYLEX • ZREVRANGEBYSCORE • ZREVRANK • ZSCAN • ZSCORE
	Sorted set-related write commands	• ZADD • ZINCRBY • ZINTERSTORE • ZPOPMAX • ZPOPMIN • ZREM • ZUNIONSTORE • BZPOPMIN • BZPOPMAX
Hash-related commands	Hash-related read commands	• HEXISTS • HGET • HLEN • HMGET • HSCAN • HSTRLEN
	Hash-related write commands	• HDEL • HINCRBY • HINCRBYFLOAT • HMESET • HSET • HSETNX

19.1.11.6.6. Data change

In DMS, you can execute SQL statements for data changes. However, the execution requires a high level of security. DMS allows you to configure security rules on the SQL Correct tab to validate the submission and approval of tickets for data changes. Only the SQL statements that are validated by the security rules can be executed.

Background information

Based on a DSL, new security rules are flexible to use. You can apply new security rules to define risk levels for tickets so that a ticket can be submitted to the approval process that is designed for the specified risk level. For more information, see DSL syntax for security rules.

Basic configuration items

Configuration item	Description
Data change default approval Template	By default, this approval template takes effect if you do not configure different approval rules for data changes at different risk levels under the Risk Approval Rules checkpoint. In the Switch Approval Template dialog box, you can change the approval process of the default approval template. For more information, see Customize approval processes.
Data Change risk level list	This risk level list defines risk levels that are used in the **Risk Identification Rules** and **Risk Approval Rules** checkpoints to identify and classify risks in data changes. You can set risk levels based on the type and scenario of data changes. Data changes at different risk levels are submitted to different approval processes. DMS allows you to set the following four risk levels: • *low*: a low risk level. • *middle*: a medium risk level. • *high*: a high risk level. • *highest*: a critical risk level.

Supported checkpoints

Note Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description

SQL execution rules	SQL execution rules are used to limit the SQL statements that can be submitted for execution. If you do not enable SQL execution rules, all SQL statements that are used for data changes cannot be executed. Assume that you want to use DML statements to change the data of a database in an online environment. You can create the following SQL execution rule:
	Example:
	`if`
	`  @fac.env_type not in ['product']`
	`  and`
	`  @fac.sql_type in [ 'UPDATE','DELETE','INSERT']`
	`then`
	`  @act.allow_submit`
	`end`
	Note:
	The preceding rule specifies that you can only submit data change tickets to execute UPDATE, DELETE, and INSERT statements on a database that is deployed in an online environment.

Checkpoint	Description
Risk Identification Rules	If a ticket conforms to the preset SQL execution rules, DMS continues to validate the ticket based on the risk identification rules. Risk identification rules are used to identify and classify risks in data changes. You can create risk identification rules based on your database environment, the number of rows in which data is affected, and the categories and subcategories of SQL statements. **Note** Different risk identification rules apply to different check items. DMS automatically identifies the highest risk level for a data change. For example, if the risk level of a data change is identified as high, medium, and low by one, three, and five risk identification rules, DMS assumes that the data change is at high risk. Example: `if @fac.env_type not in ['product','pre'] then @act.mark_risk 'low 'Low risk level: offline environment' end` Note: The preceding rule specifies that if the destination database is deployed in an offline environment, data changes are at low risk.
Risk Approval Rules	After the risk level of a data change is identified by the risk identification rules, DMS processes the ticket based on the risk approval rules. You can customize risk approval rules under the **Risk Approval Rules** checkpoint. **Note** • If a data change does not hit risk approval rules, DMS uses the default approval template that is specified under the Basic Configuration Item checkpoint to process the ticket. • By default, an offline environment is identified as a factor at low risk and requires no approval.
Batch Data import rules	These rules apply only to the validation of data import tickets. You can use the default rules that are provided in templates, or configure rules based on your actual needs.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`.The following table describes the supported factors on the SQL Correct tab.

Factor	Description
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.
@fac.sql_type	The type of the SQL statement. The value is the subcategory of the SQL statement, such as UPDATE or INSERT. For more information, see Supported SQL statements.

Factor	Description
@fac.detail_type	The type of the data change. Valid values: • *COMMON*: a Normal Data Modify ticket. • *CHUNK_DML*: a Lock-Free Data Modify ticket. • *PROCEDURE*: a Programmable Object ticket. • *CRON_CLEAR_DATA*: a History Data Clean ticket. • *BIG_FILE*: a Large Data Import ticket.
@fac.is_logic	A Boolean value that indicates whether the database to be affected is a logical database.
@fac.extra_info	The additional information about the data change. This factor is not in use.
@fac.is_ignore_affect_rows	A Boolean value that indicates whether to skip the validation.
@fac.insert_rows	The number of rows of data to be inserted.
@fac.update_delete_rows	The number of rows of data to be updated.
@fac.max_alter_table_size	The size of the largest tablespace where the table to be modified is stored.
@fac.is_has_security_column	A Boolean value that indicates whether the SQL statement to be executed involves sensitive fields.
@fac.security_column_list	A list of sensitive fields that the SQL statement to be executed involves.
@fac.risk_level	The risk level of the operation that is to be performed by the SQL statement.
@fac.risk_reason	The reason for identifying the operation to be performed as at the risk level.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported action on the SQL Correct tab.

Action	Description
@act.allow_submit	Requires the submission of SQL statements to be executed in a ticket.
@act.allow_execute_direct	Allows the execution of SQL statements in the SQLConsole.
@act.forbid_execute	Forbids the execution of SQL statements.
@act.mark_risk	Marks the risk level of a data change. Example: `@act.mark_risk 'Medium risk level: online environment'`.
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	

Action	Description
@act.choose_approve_template_with_reason	

19.1.11.6.7. Permission application

DMS allows you to configure security rules on the Access apply tab to validate applications for permissions, including permissions on instances, databases, and tables.

Background information

In DMS, security rules are flexible to use. You can apply security rules to define risk levels for tickets so that a ticket can be submitted to the approval process that is designed for the specified risk level. For more information about the DSL syntax, see DSL syntax for security rules.

Basic configuration items

The following table describes the basic configuration items that are supported on the Access apply tab.

Configuration item	Description
[Instance-permission application] default approval Template	By default, this approval template takes effect if you do not set different approval processes for instance permission applications at different risk levels under the **Validation for Instance Permission Application** checkpoint. **Note** In the Switch Approval Template dialog box, you can change the approval process of the default approval template.
[DB-permission application] default approval Template	By default, this approval template takes effect if you do not set different approval processes for database permission applications at different risk levels under the **Database Permission Application Validation** checkpoint.
Table-permission request default approval Template	By default, this approval template takes effect if you do not set different approval processes for table permission applications at different risk levels under the **Table Permission Application Validation** checkpoint.
[Programmable object-permission application] default approval Template	By default, this approval template takes effect if you do not set different approval processes for programmable object permission applications at different risk levels under the **Programmable object verification** checkpoint.
[Field-permission application] default approval Template	By default, this approval template takes effect if you do not set different approval processes for sensitive field permission applications at different risk levels under the **Sensitive Field Application Validation** checkpoint.
Line-permission application default approval Template	By default, this approval template takes effect if you do not set different approval processes for row permission applications at different risk levels under the **Line permission application verification** checkpoint.
[Owner-application] default approval template (when the resource has no Owner)	By default, this approval template takes effect if you do not set different approval processes for data ownership applications at different risk levels under the **Owner Application Validation** checkpoint and the data that is involved in the application has no owner.

Configuration item	Description
[Owner-application] default approval template (when the resource has an Owner)	By default, this approval template takes effect if you do not set different approval processes for data ownership applications at different risk levels under the **Owner Application Validation** checkpoint and the data that is involved in the application has one or more owners.

Supported checkpoints

When a user submits a ticket to apply for permissions, DMS checks whether the ticket conforms to rules that are specified under checkpoints. The ticket can be submitted only after DMS determines that the ticket conforms to all rules that are specified under checkpoints.

> **Note** Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description
Owner Application Validation	Allows you to set approval processes or constraints for **Instance-OWNER**, **Table-OWNER**, and **Database-OWNER** tickets.
Validation for Instance Permission Application	Allows you to set approval processes or constraints for **Instance-Performance** and **Instance-Login** tickets.
Database Permission Application Validation	Allows you to set approval processes or constraints for **Database-Permission** tickets.
Table Permission Application Validation	Allows you to set approval processes or constraints for **Table-Permission** tickets.
Programmable object verification	Allows you to set approval processes or constraints for **Programmable Object** tickets.
Sensitive Field Application Validation	Allows you to set approval processes or constraints for **Sensitive Column-Permission** tickets.
Line permission application verification	Allows you to set approval processes or constraints for **Row-Permission** tickets.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and database names. A factor name starts with the prefix `@fac.` .The following table describes the supported factors on the Access apply tab.

Factor	Description
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.
@fac.schema_name	The name of the database.
@fac.perm_apply_duration	The period of time during which the applicant needs the permission. Unit: hours.

Factor	Description
@fac.column_security_level	The security level of the field. Valid values: • *sensitive* • *confidential* • *inner*

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported action on the Access apply tab.

Action	Description
@act.forbid_submit_order	Forbids a ticket from being submitted.
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	
@act.choose_approve_template_with_reason	

19.1.11.6.8. Data export

DMS allows you to manage security rules on the Data Export tab to validate the permissions of applicants on involved databases, tables, sensitive fields, and rows during the submission and approval of data export tickets. This helps ensure data security.

Basic configuration items

Data export default approval Template: the default approval template that takes effect if you do not set different approval processes for data export tickets at different risk levels under the Approval Rule Validation checkpoint. In the Switch Approval Template dialog box, you can change the approval process of the default approval template. For more information, see Customize approval processes.

Supported checkpoints

Note Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description
Pre-check Validation	Allows you to specify whether to validate the permissions of applicants on involved databases, tables, sensitive fields, and rows by configuring security rules.
Approval Rule Validation	Allows you to submit data export tickets to different approval processes by configuring security rules. For example, you can submit tickets for exporting more than a specific number of rows to an approval process and other tickets to another approval process.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`. The following table describes the supported factors on the Data Export tab.

Factor	Description
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.
@fac.is_ignore_export_rows_check	A Boolean value that indicates whether to skip the check on the number of rows to be affected.
@fac.export_rows	The number of rows to be exported.
@fac.include_sec_columns	A Boolean value that indicates whether the data to be exported contains sensitive fields.
@fac.sec_columns_list	The sensitive fields that require or do not require approval before data is exported. The sensitive fields are displayed in the format of `Table name.Field name, [Table name.Field name, ...]`.
@fac.user_is_admin	A Boolean value that indicates whether the applicant is a DMS administrator.
@fac.user_is_dba	A Boolean value that indicates whether the applicant is a DBA.
@fac.user_is_inst_dba	A Boolean value that indicates whether the applicant is the DBA of the current instance.
@fac.user_is_sec_admin	A Boolean value that indicates whether the applicant is a security administrator.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`. The following table describes the supported actions on the Data Export tab.

Action	Description
@act.do_not_approve	Allows a ticket to be processed without approval.
@act.choose_approve_template	Specifies an approval template.
@act.choose_approve_template_with_reason	Specifies an approval template with a reason provided.
@act.forbid_submit_order	Forbids a ticket from being submitted.
@act.enable_check_permission	Validates the permissions of an applicant on involved databases and tables.
@act.disable_check_permission	Does not validate the permissions of an applicant on involved databases and tables.
@act.enable_check_sec_column	Validates the permissions of an applicant on involved sensitive fields.

Action	Description
@act.disable_check_sec_column	Does not validate the permissions of an applicant on involved sensitive fields.

19.1.11.6.9. Schema design

DMS allows you to configure security rules on the Schema Design tab to check the design rules and risk identification rules that apply to schema design tickets. This helps ensure data security.

Basic configuration items

Configuration item	Description
Enable non-peer Publishing	Specifies whether to enable non-peer publishing. By default, data changes to a table can be published only to a table with the same name in another database. After you enable non-peer publishing, you can perform data changes on all tables. **Warning** This feature may bring high risks. We recommend that you proceed with caution and enable this feature only for special requirements.
R & D process	The whole process of a schema design ticket. It is the most important configuration item on the Schema Design tab. For more information about the parameters of the configuration item, see Parameters involved in the R&D process.
Field type configuration	The supported data types of fields to be added.
Index type configuration	The supported data types of indexes to be added.
It is forbidden to modify the original field data type	Specifies whether to prohibit the data types of the original fields from being modified when the original table is to be modified.
Prohibit deleting original fields	Specifies whether to prohibit the existing fields from being deleted when the original table is to be modified. **Note** We recommend that you enable this feature because deleting existing fields may bring high risks.
Prohibit renaming original fields	Specifies whether to prohibit the existing fields from being renamed when the original table is to be modified. **Note** We recommend that you enable this feature because reaming existing fields may bring high risks.
Table character set license configuration	The range of character sets that are allowed to be used when you create a table. For example, you can specify utf8 and utf8mb4.
Default approval template for Structural design	The default approval template that is used for a schema design ticket if you do not configure the Approval Rule Validation checkpoint. In the Switch Approval Template dialog box, you can change the approval process of the default approval template.

Configuration item	Description
When published, the ticket will automatically advance to the end state	The point that is used to stop the schema change process. If you enable this feature, after the node that is set as the anchor in the R&D process is run, DMS automatically turns the ticket to the Finished state. **Note** To use this feature, you must set the last node in the R&D process as the anchor.

Supported checkpoints

Note Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

The Schema Design tab contains the following two processes:

- Process of saving changes: DMS provides the following three checkpoints for this process. The checkpoints validate the table headings, fields, and indexes.
  - **Save Changes and Validate Header**
  - **Save Changes and Validate Field**
  - **Save Changes and Validate Index**
- Process of applying changes: DMS provides the following five checkpoints for this process. The first four checkpoints identify the risks that arise from changing schemas without locking tables, and the last checkpoint assigns an approval process to each type of risk.
  - **Table Creation Risk Control**
  - **Field Change Risk Control**
  - **Index Change Risk Control**
  - **SQL Execution Risk Control**
  - **Approval Rule Validation**

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`. The following table describes the supported factors on the Schema Design tab.

Factor	Description
@fac.table_kind	The type of the table whose schema is to be changed. Valid values: • *new*: a newly created table. • *old*: an existing table.
@fac.column_kind	The type of the field to be changed. Valid values: • *new*: a newly created field. • *old*: an existing field.
@fac.xxxx_old	The value of an existing field or index that is used for comparison.
@fac.column_is_primary	A Boolean value that indicates whether the current field serves as a primary key. Valid values: *true* and *false*.

Factor	Description
@fac.column_type_support_default	A Boolean value that indicates whether the data type of the current field supports a default value. Valid values: *true* and *false*. **Note** For example, a field of the CHAR type supports a default value, whereas a field of the TEXT type does not.
@fac.index_kind	The type of the index to be changed. Valid values: • *new*: a newly created index. • *old*: an existing index.
@fac.index_column_count	The number of fields in the index.
@fac.change_type	The type of the schema change to be performed by DDL statements. Valid values: • *add*: adds one or more fields or indexes. • *modify*: modifies one or more fields or indexes. • *delete*: deletes one or more fields or indexes.
@fac.altered_table_size	The size of the table whose schema is to be changed. Unit: MB.
@fac.online_execute	A Boolean value that indicates whether the schema change can be performed in an online environment. Valid values: *true* and *false*.
@fac.change_risk_level	The risk level of the schema change. Valid values: • *high*: a high risk level. • *middle*: a medium risk level. • *low*: a low risk level.
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions on the Schema Design tab.

Action	Description	Format
@act.block_submit	Blocks the submission of the schema change and displays the error message. This action can be used in the process of saving changes.	@act.block_submit 'Reason for blocking the submission'
@act.show_warning	Displays the error message without blocking the submission of the schema change. This action can be used in the process of saving changes.	@act.show_warning 'Error message'

Action	Description	Format
@act.mark_middle_risk	Specifies that the schema change is at medium risk. This action can be used in the process of identifying the risk level.	@act.mark_middle_risk 'Reason for the identification'
@act.mark_high_risk	Specifies that the schema change is at high risk. This action can be used in the process of identifying the risk level.	@act.mark_high_risk 'Reason for the identification'
@act.forbid_submit_publish	Rejects the ticket. This action can be used in the process of setting the approval process.	@act.forbid_submit_publish 'Reason for the rejection'
@act.do_not_approve	Specifies the ID of an approval template.	N/A
@act.choose_approve_template		
@act.choose_approve_template_with_reason		

Parameters involved in the R&D process

Parameter	Description
Step	• The type of the node. Valid values: • **Design**: The design node in the R&D process is generated by default and cannot be removed. It determines the environment where the schema change is designed. • **Publish**: A publish node in the R&D process is used to publish the schema change after the change is designed. You can set multiple publish nodes.
Node Name	The name of the node. The node name can be up to 10 characters in length.
Database Environment	The environment where the node is run.
Execution Strategy	• The way in which the node is run. Valid values: • **Immediately**: The node is run immediately after it is approved. • **Periodically**: The node is run at the time that you specify. **Note** If a node is approved before the specified point in time, it is run as scheduled. Otherwise, the node is interrupted and not run.
Can Go Back	Specifies whether a publish node can be rolled back to the design node.
Can Skip	Specifies whether the current node can be skipped.
Anchor	The point that is used to stop the schema change process. If you set a node as the anchor, after the node is published, the nodes that follow the anchor cannot be run and the schema change process ends. At this time, the ticket enters the Published state.
Actions	The operation that you can perform on a publish node. You can remove a publish node as required.

19.1.11.6.10. Database and table synchronization

DMS allows you to configure security rules on the Table Sync tab to validate operations that are related to schema synchronization, empty database initialization, and table consistency repair.

Basic configuration items

Configuration item	Description
Enable execution capability	Specifies whether to enable SQL-based synchronization. If this configuration item is set to OFF, applicants can compare table schemas but cannot execute SQL statements to synchronize databases and tables. Other configuration items and security rules you set under checkpoints on the Table Sync tab also become invalid.
Database table synchronization default approval Template	The default approval template for database and table synchronization applications. You can use the default approval template or click Switch Approval Template and select another template. For more information, see Customize approval processes.
Analysis phase script Expiration Time (unit: hours)	The timeout period of the analysis phase. You can set an appropriate timeout period in which synchronization can be canceled if schemas are changed in the destination database.

Supported checkpoints

The Table Sync tab contains three checkpoints that are corresponding to the three features that are supported by the tab. The three checkpoints are unrelated to each other. For example, when you submit a Schema Synchronization ticket, only the basic configuration items and the security rules that are specified under the Schema Synchronization Validation checkpoint are used to validate the ticket.

> **Note** Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description
Schema Synchronization Validation	Allows you to set approval processes or constraints for Schema Synchronization tickets.
Empty Database Initialization Validation	Allows you to set approval processes or constraints for Empty Database Initialization tickets.
Table Consistency Repair Validation	Allows you to set approval processes or constraints for Repair Table Consistency tickets.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`.The following table describes the supported factors on the Table Sync tab.

Factor	Description
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.
@fac.schema_name	The name of the schema.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions on the Table Sync tab.

Action	Description
@act.forbid_submit_order	Forbids a ticket from being submitted. The statement is in the following format: `@act.forbid_submit_order 'Reason for forbidding the submission of the ticket'`.
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	
@act.choose_approve_template_with_reason	

19.1.11.6.11. Sensitive field change

The topic describes the security rules on the Sensitive Column Change tab.

Basic configuration items

Sensitive column default approval Template: the default approval template that takes effect if you do not set approval processes for tickets that apply to change the security levels of sensitive fields under the Approval Rule Validation checkpoint. In the Switch Approval Template dialog box, you can change the approval process of the default approval template.

Supported checkpoints

Approval Rule Validation: When a user submits a ticket to change the security level of a sensitive field, DMS checks whether the ticket conforms to the rules that are specified under the Approval Rule Validation checkpoint.

> **Note** Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`.The following table describes the supported factors on the Sensitive Column Change tab.

Factor	Description

@fac.column_level_change_type	The type of security level change that the applicant wants to perform on a sensitive field. Valid values:
	• *upper*: raises the current security level, including the following three cases:
	  ◦ From inner to sensitive
	  ◦ From inner to confidential
	  ◦ From sensitive to confidential
	• *sensitive_to_inner*: lowers the security level from sensitive to inner.
	• *confidential_to_sensitive*: lowers the security level from confidential to sensitive.
	• *confidential_to_inner*: lowers the security level from confidential to inner.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions on the Sensitive Column Change tab.

Action	Description
@act.forbid_submit_order	Forbids a ticket from being submitted. The statement is in the following format: `@act.forbid_submit_order 'Reason for forbidding the submission of the ticket'`.
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	
@act.choose_approve_template_with_reason	

19.1.11.6.12. Test data generation

This topic describes the security rules on the Test Data Generate tab.

Supported checkpoints

Approval rule validation: When a user submits a ticket to generate test data, DMS checks whether the ticket conforms to the rules that are specified under the Approval rule validation checkpoint.

> **Note** Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`.The following table describes the supported factors on the Test Data Generate tab.

Factor	Description
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.
@fac.schema_name	The name of the schema.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions on the Test Data Generate tab.

Action	Description

Action	Description
@act.forbid_submit_order	Forbids a ticket from being submitted. The statement is in the following format: `@act.forbid_submit_order 'Reason for forbidding the submission of the ticket'` .
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	
@act.choose_approve_template_with_reason	

19.1.11.6.13. Database cloning

This topic describes the security rules on the Database Clone tab.

Basic configuration items

Database clone default approval Template: the default approval template that takes effect if you do not set approval processes for database clone tickets under the Approval rule validation checkpoint. In the Switch Approval Template dialog box, you can change the approval process of the default approval template.

Supported checkpoints

Approval rule validation: When a user submits a database clone ticket, DMS checks whether the ticket conforms to the rules that are specified under the Approval rule validation checkpoint.

> **Note** Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.` .The following table describes the supported actions on the Database Clone tab.

Action	Description
@act.forbid_submit_order	Forbids a ticket from being submitted. The statement is in the following format: `@act.forbid_submit_order 'Reason for forbidding the submission of the ticket'` .
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	
@act.choose_approve_template_with_reason	

19.1.11.7. Configuration management

DMS allows DMS administrators to manage system configurations. If you are a DMS administrator, you can modify the system configuration items to flexibly meet your business requirements.

Prerequisites

You are a DMS administrator.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **System Management > Configuration**.
3. Find the parameter that you want to modify and click **Change** in the **Actions** column.

   > **Note** You can also click **Change History** to view the change history of the parameter.

4. In the Change Parameter Configuration dialog box, enter a specific value.
5. Click **Confirm Change**.

Types of data changes

key	value	Description
config_correct	Modify Config	To modify configurations.
project_init_data	Init Project Data	To initialize the data for a project.
program_bug	Program Bug	To fix a bug.
require_deal_without_backend_function	Requirements Without Backend Function	To manage the data of an application that does not support backend management.
history_data_clear	History Data Clean	To clear historical data.
test	Test	To run a test.
mis_operation	Mis Operation	To restore data after a misoperation.
others	Others	To change data for other reasons.

20.Server Load Balancer (SLB)

20.1. User Guide

20.1.1. What is SLB?

This topic provides an overview of Server Load Balancer (SLB). SLB distributes inbound network traffic across multiple Elastic Compute Service (ECS) instances that act as backend servers based on forwarding rules. You can use SLB to improve the responsiveness and availability of your applications.

Overview

After you add ECS instances that are deployed in the same region to a SLB instance, SLB uses virtual IP addresses (VIPs) to virtualize these ECS instances into backend servers in a high-performance server pool that ensures high availability. Client requests are distributed to the ECS instances based on forwarding rules.

SLB checks the health status of the ECS instances and automatically removes unhealthy ones from the server pool to eliminate single points of failure (SPOFs). This enhances the resilience of your applications.

Components

SLB consists of three components:

- SLB instances

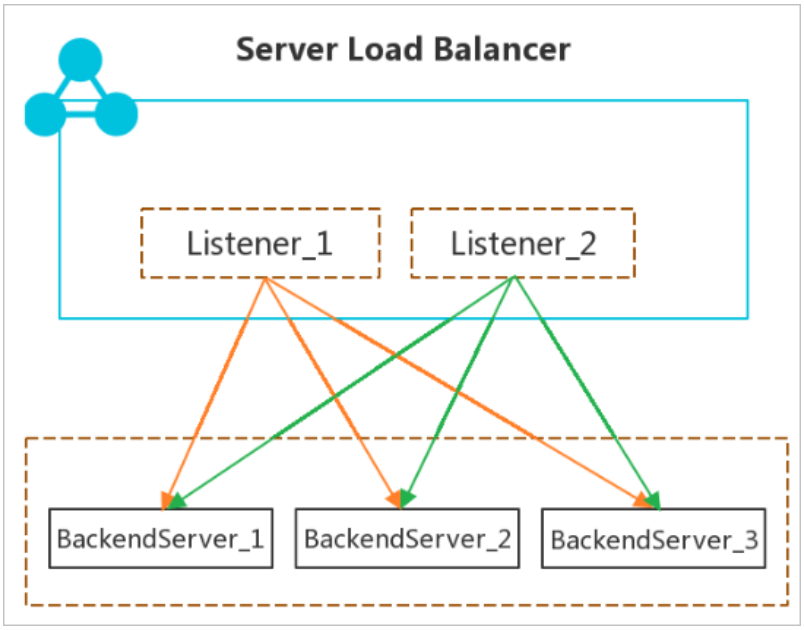
  A SLB instance is a running SLB service entity that receives traffic and distributes traffic to backend servers. To get started with SLB, you must create a SLB instance and add at least one listener and two ECS instances to the SLB instance.

- Listeners

  A listener checks client requests and forwards them to backend servers. It also performs health checks on backend servers.

- Backend servers

  ECS instances are used as backend servers to receive distributed requests. You can separately add ECS instances to the server pool, or use vServer groups or primary/secondary server groups to add and manage ECS instances in batches.



Benefits

- High availability

  SLB is designed with full redundancy that avoids SPOFs and supports zone-disaster recovery.

  SLB can be scaled based on application loads and can provide continuous service during traffic fluctuations.

- High scalability

  You can increase or decrease the number of backend servers to adjust the load balancing capability of your applications.

- Cost-effectiveness

  SLB can save 60% of load balancing costs compared with using traditional hardware solutions.

- Security

  You can use SLB with Apsara Stack Security to defend your applications against up to 5 Gbit/s DDoS attacks.

- High concurrency

  A SLB cluster supports hundreds of millions of concurrent connections and a single SLB instance supports tens of millions of concurrent connections.

20.1.2. Log on to the SLB console

This topic describes how to go to the Server Load Balancer (SLB) console after you log on to the Apsara Uni-manager Management Console by using the Chrome browser.

Prerequisites

- The domain name of the Apsara Uni-manager Management Console is obtained from the engineer that deploys the service before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Login**.
4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations in the following scenarios:
   - It is the first time that you log on to the console after MFA is forcibly enabled by the administrator.
     a. On the Bind Virtual MFA Device page, bind an MFA device.
     b. Enter the account and password again as in Step 2 and click **Log On**.

c. Enter a six-digit MFA verification code and click **Authenticate**.

- You have enabled MFA and bound an MFA device.

  Enter a six-digit MFA authentication code and click **Authenticate**.

  > **Note** For more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Operations Console User Guide*.

5. In the top navigation bar, choose **Products > Networking > Server Load Balancer**.

20.1.3. Quick start

20.1.3.1. Overview

This quick start tutotial describes how to create a public-facing Server Load Balancer (SLB) instance and how to forward requests to two backend servers.

> **Note** Before creating an SLB instance, you must determine the region, type, and billing method of the SLB instance. For more information, see Preparations.

This tutotial includes the following content:

1. Create a CLB instance

   Create an SLB instance. An SLB instance is a running entity of the SLB service.

2. Add listeners and backend servers.

   Configure listening rules and backend servers for the SLB instance.

3. Release an SLB instance

   If you no longer need the SLB instance, delete it to avoid extra fees.

20.1.3.2. Before you begin

This article presents the essential considerations for configuring an SLB instance. Before you create an SLB instance, you must determine the types of listeners and the network traffic you want to balance.

Instance region

When you select a region, note the following points:

- To reduce latency and increase the download speed, we recommend that you select a region closest to your end-users.
- SLB offers stable and reliable load balancing services by providing support for primary/secondary failovers in most regions. This implements disaster recovery across different zones within the same region. We recommend that you select a region that supports the primary/secondary SLB deployment.
- SLB instances cannot span across regions. Therefore, you must make sure that the SLB instance and its backend Elastic Compute Service (ECS) instances are located in the same region.

Network traffic

SLB provides load balancing services for both Internet and internal network traffic:

- If you need to use SLB to distribute requests from the Internet, you can create an Internet-facing SLB instance.

  An Internet-facing SLB instance comes with a public IP address to receive requests from the Internet.

- If you need to use SLB to distribute requests from the internal network, you can create an internal SLB instance.

Internal SLB instances only have private IP addresses and are only accessible from the internal network and not from the Internet.

Listener protocol

SLB supports Layer-4 load balancing of Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) traffic, and Layer-7 load balancing of HTTP and HTTPS traffic.

- A Layer-4 listener directly distributes requests to backend servers without modifying packet headers. After a client request reaches a Layer-4 listener, SLB uses the backend port configured for the listener to establish a TCP connection with an Elastic Compute Service (ECS) instance (backend server).
- A Layer-7 listener is implemented as a reverse proxy. After a client request reaches a Layer-7 listener, SLB establishes a new TCP connection over HTTP to a backend server, instead of directly forwarding the request to the backend server (ECS instance).

  Compared with Layer-4 listeners, Layer-7 listeners require an additional step of Tengine processing. Therefore, Layer-4 listeners provide better performance than Layer-7 listeners. In addition, the performance of Layer-7 listeners can also be affected by factors such as insufficient client ports or excessive backend server connections. Therefore, we recommend that you use Layer-4 listeners for high-performance load-balancing services.

Backend servers

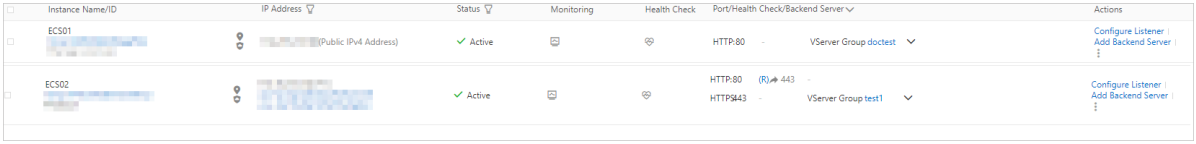
Before you use the SLB service, you must create ECS instances, deploy applications on them, and add the ECS instances to your SLB instance to process client requests.

When you create and configure an ECS instance, note the following points:

- Select a region and zone for the ECS instance

  Make sure that the ECS instance resides in the same region and Virtual Private Cloud (VPC) as the SLB instance. We recommend that you deploy ECS instances in different zones to improve availability.

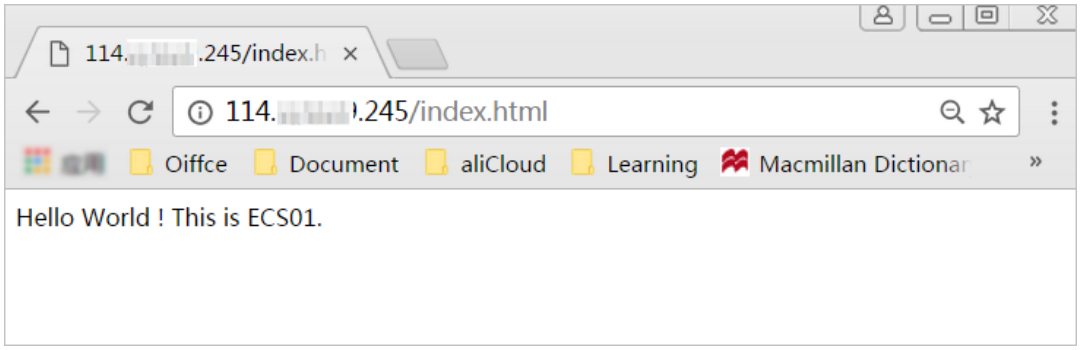
  In this example, two ECS instances named ECS01 and ECS02 are created in the **China (Hangzhou)** region. The following figure shows their basic configurations.



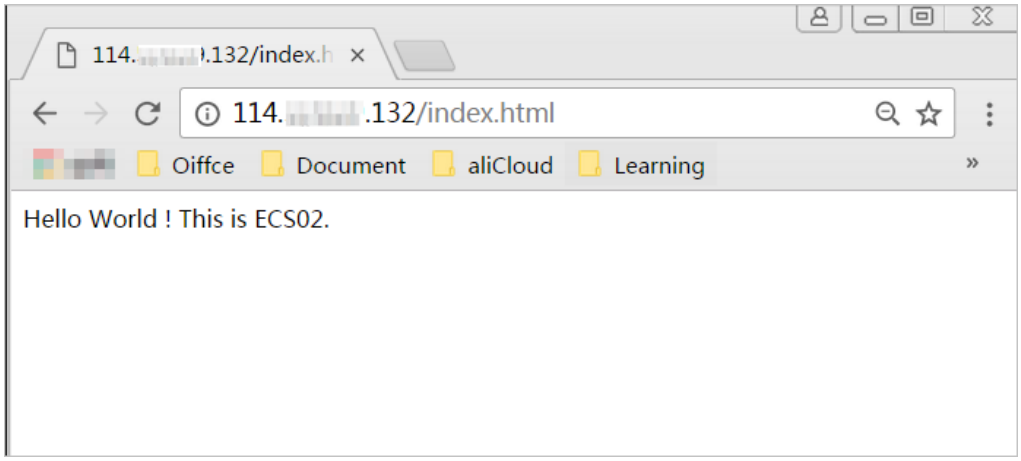
- Configure applications

  In this example, two static web pages are built on ECS01 and ECS02 by using Apache.

  - Enter the Elastic IP address (EIP) associated with ECS01 in the address box of your browser.



- Enter the EIP associated with ECS02 in the address box.



No additional configuration is required after you deploy applications on the ECS instances. However, if you need to use a Layer-4 (TCP or UDP) listener and the ECS instances run on Linux, make sure that the following parameters in the *net.ipv4.conf* file under */etc/sysctl.conf* are set to 0:

```
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.eth0.rp_filter = 0
```

20.1.3.3. Create an SLB instance

This topic describes how to create a Server Load Balancer (SLB) instance. An SLB instance is a running entity of the SLB service. You can add multiple listeners and backend servers to an SLB instance.

Prerequisites

- Elastic Compute Service (ECS) instances are created and applications are deployed on the ECS instances.
- The ECS instances and the SLB instance belong to the same organization. In addition, the security group rules of the ECS instances allow access from port 80 (HTTP) and port 443 (HTTPS).

Procedure

1. Log on to the SLB console.
2. In the left-side navigation pane, choose **Instances > Instances**.
3. On the **Instances** page, click **Create Instance**.
4. Configure the SLB instance and click **Submit**.

Parameter	Description
Organization	Select an organization for the SLB instance from the drop-down list. **Note** Make sure that the organization of the SLB instance is the same as that of its backend servers.
Resource Set	Select the region where you want to deploy the SLB instance.
Region	Select the region where you want to deploy the SLB instance.

Parameter	Description
Zone	Select a zone for the SLB instance from the drop-down list.
Instance Name	Enter a name for the SLB instance in the Instance Name field. The name must be 2 to 128 characters in length, and can contain letters, digits, full-width characters, underscores (_), hyphens (-), periods (.), and colons (:). Line breaks and spaces are supported. The name must start with a letter and cannot start with `http://` or `https://`.
Instance Edition	Select one of the following options: ○ Shared-performance: Shared-performance SLB instances share resources with each other. The performance of shared-performance SLB instances is not guaranteed. ○ Guaranteed-performance: Guaranteed-performance SLB instances use exclusive resources. The performance of guaranteed-performance SLB instances varies by type.
Instance Type	Select the type of network traffic that you want to distribute. Valid values: Internal Network and Internet. Internal Network is selected in this example.
Network Type	Select the network type of the SLB instance. Valid values: Classic Network and VPC. VPC is selected in this example.
VPC	Select a virtual private cloud (VPC).
vSwitch	Select a vSwitch.
IP Version	Select an IP version.
IP Address	Enter a service IP address. Make sure that the service IP address is valid. Otherwise, the SLB instance cannot be created. If you do not set this parameter, the system automatically allocates an IP address to the SLB instance. **Note** The private IP address that you specify must belong to the destination CIDR block of the vSwitch.

What's next

Configure a CLB instance

20.1.3.4. Configure an SLB instance

This topic describes how to configure a Server Load Balancer (SLB) instance. After you create an SLB instance, you must add at least one listener and one group of backend servers to the SLB instance so that it can forward traffic. The following example describes how to add a TCP listener and two Elastic Compute Service (ECS) instances to an SLB instance as backend servers. The ECS instances are ECS01 and ECS02 that host static web pages.

Procedure

1. Log on to the SLB console
2. On the **Instances** page, find the SLB instance that you want to manage and click **Configure Listener** in the Actions column.

3. On the **Protocol and Listener** wizard page, set the required parameters based on the following information. Use the default settings for other parameters.
   - **Select Listener Protocol**: Select a listener protocol. **TCP** is selected in this example.
   - **Listening Port**: Specify a frontend port to receive and forward requests to backend servers.

     The SLB instance uses this port to provide external services. In most cases, port 80 is set for HTTP listeners and port 443 is set for HTTPS listeners.

     This parameter is set to **80** in this example.

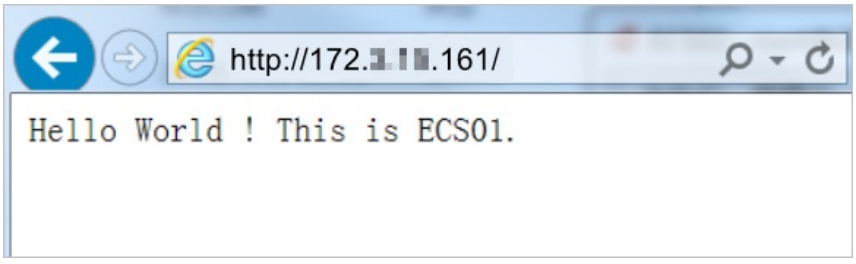
   Advanced:

   - **Enable Peak Bandwidth Limit**: Applications running on backend ECS instances provide external services. You can set bandwidth caps to limit service capabilities of applications.
   - **Scheduling Algorithm**: SLB supports the following three scheduling algorithms. **Round-Robin (RR)** is selected in this example.
     - Weighted Round-Robin (WRR): Requests are sequentially distributed to backend servers. Backend servers with higher weights receive more requests.
     - Weighted Least Connections (WLC): In addition to the weight of each backend server, the number of connections of each backend server also affects request distribution. If multiple backend servers have the same weight, requests are routed to the backend server with the least connections.
     - Round-Robin (RR): Requests are evenly and sequentially distributed to backend servers.
4. Click **Next**. On the **Backend Servers** wizard page, select **Default Server Group** and click **Add More** to add backend servers.
   i. In the **My Servers** panel, select the created ECS instances, ECS01 and ECS02, and click **Next**.
   ii. A backend server with a higher weight receives more requests. The default value is 100. We recommend that you use the default value.
   iii. Click **Add**.
   iv. On the **Default Server Group** tab, specify backend ports. The ports are used by backend ECS instances to receive requests. You can specify the same port for multiple backend servers that are added to the same SLB instance. The port is set to 80 in this example.
5. Click **Next** to configure the health check feature. The default health check settings are used in this example.

   After you enable the health check feature, when an ECS instance is considered unhealthy, SLB sends new requests to other healthy ECS instances. SLB only sends requests to this ECS instance after it is recovered and considered healthy.
6. Click **Next**. On the **Confirm** wizard page, check the configurations and click **Submit**.
7. Click **OK** to go back to the **Instances** page and click to refresh the page.

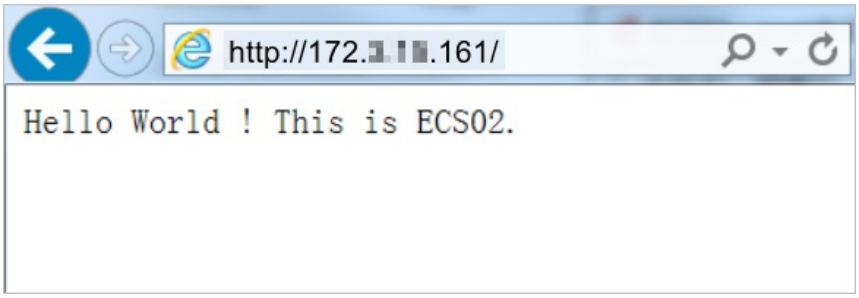
   If the health check state of a backend ECS instance is **Active**, the backend server is working as expected and can process requests.
8. Enter the service address of the SLB instance into the address bar of the browser to test load balancing services of the instance.

   ECS01

   ECS02

ECS02



20.1.3.5. Release an SLB instance

If an SLB instance is no longer needed, you can release the instance to save costs. The backend ECS instances will not be deleted or affected after you delete an SLB instance.

Procedure

1. Log on to the SLB console.
2. On the **Instances** page, find the instance and click > **Release** in the Actions column, or select the instance and click **Release** at the lower part of the page.
3. In the **Release** dialog box, select **Release Now**.

   > **Note** The system performs release operations at 30-minute and hour marks. However, billing for the SLB instance is stopped at the specified release time.

4. Click **Next**.
5. Click **OK** to release the SLB instance.

   > **Note** Pay-as-you-go SLB instances cannot be restored once deleted. We recommend that you exercise caution when you release SLB instances.

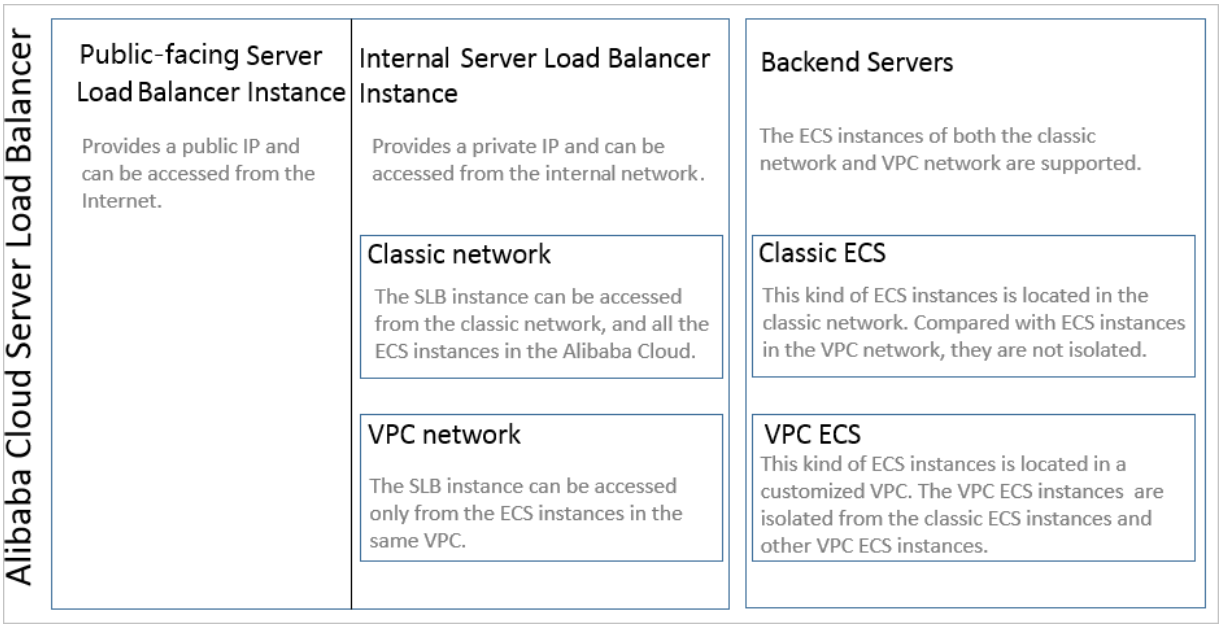
20.1.4. SLB instances

20.1.4.1. SLB instance overview

A Server Load Balancer (SLB) instance is a running entity of the SLB service. To use the SLB service, you must create an SLB instance and add listeners and backend servers to the instance.

Instance network types

Apsara Stack provides Internet-facing and internal-facing SLB instances. You can create Internet-facing SLB instances or internal-facing SLB instances based on your business requirements. If you create an Internet-facing SLB instance, a public IP address is allocated. If you create an internal-facing SLB instance, a private IP address is allocated.



- Internet-facing SLB instances

  An Internet-facing SLB instance distributes client requests over the Internet to backend servers based on configured forwarding rules. An internal-facing SLB instance in a virtual private cloud (VPC) can process requests sent over the Internet only when the instance is associated with an elastic IP address (EIP). The following table provides more detailed information.

Type	Feature
Internet-facing SLB instances After you create an Internet-facing SLB instance, the system allocates a public IP address to the instance. You can bind a domain name to the public IP address to provide external services.	○ The public IP address is allocated to the SLB instance and cannot be unbound from the instance. ○ Internet-facing SLB instances support only pay-by-data-transfer and pay-by-bandwidth billing methods.
Internal-facing SLB instances that are associated with EIPs An internal-facing SLB instance that is associated with an EIP can process requests sent over the Internet.	○ A public IP address is allocated to the EIP. You can associate the EIP with the SLB instance and disassociate the EIP from the SLB instance based on your requirements. ○ An EIP that is associated with an EIP bandwidth plan supports the 95th percentile bandwidth billing method.

- Internal-facing SLB instances

  Internal-facing SLB instances can be used only inside Apsara Stack and can forward only requests from clients that can access SLB instances over the internal network.

  You can select one of the following network types for an internal-facing SLB instance:

- Classic network

  If you choose classic network for an internal-facing SLB instance, the IP address of the SLB instance is allocated and maintained by Apsara Stack. This instance can be accessed only by Elastic Compute Service (ECS) instances in the classic network.

- VPC

  If you choose VPC for an internal-facing SLB instance, the IP address of the SLB instance is allocated from the CIDR block of the vSwitch that is attached to the VPC. This SLB instance can be accessed only by ECS instances in the VPC.

Instance types and specifications

Alibaba Cloud provides shared-performance SLB instances and guaranteed-performance SLB instances. Guaranteed-performance SLB instances provide reliable performance metrics.

- Shared-performance SLB instances

  All shared-performance SLB instances share SLB resources. This indicates that the instance performance cannot be guaranteed.

- Guaranteed-performance SLB instances

  The following content describes three key metrics of guaranteed-performance SLB instances:

Apsara Stack provides four types of guaranteed-performance SLB instances.

Type	Specification	Max connection	CPS	QPS	Purchase method
Type 1	Small I (slb.s1.small)	5,000	3,000	1,000	Available for purchase from the official website of Apsara Stack.
Type 2	Standard I (slb.s2.small)	50,000	5,000	5,000	Available for purchase from the official website of Apsara Stack.
Type 3	Standard II (slb.s2.medium)	100,000	10,000	10,000	Available for purchase from the official website of Apsara Stack.
Type 4	Higher I (slb.s3.small)	200,000	20,000	20,000	Available for purchase from the official website of Apsara Stack.

The following table describes the differences between shared-performance SLB instances and guaranteed-performance SLB instances.

Feature	Shared-performance SLB instance	Guaranteed-performance SLB instance
Resource allocation	Shared resources	Exclusive resources

Feature	Shared-performance SLB instance	Guaranteed-performance SLB instance
Service level agreement for guaranteed availability	Not supported	99.95%
IPv6	×	√
Server Name Indication (SNI) certificates	×	√
Support for blacklists and whitelists	×	√
Elastic network interface (ENI) mounting	×	√
Assignment of secondary IP addresses to ENIs that are bound to ECS instances	×	√
HTTP-to-HTTPS redirection	×	√
Consistent hashing	×	√
TLS security policies	×	√
HTTP2	×	√
Websocket(S)	×	√

20.1.4.2. Create an SLB instance

This topic describes how to create a Server Load Balancer (SLB) instance. An SLB instance is a running entity of the SLB service. You can add multiple listeners and backend servers to an SLB instance.

Prerequisites

- Elastic Compute Service (ECS) instances are created and applications are deployed on the ECS instances.
- The ECS instances and the SLB instance belong to the same organization. In addition, the security group rules of the ECS instances allow access from port 80 (HTTP) and port 443 (HTTPS).

Procedure

1. Log on to the SLB console.
2. In the left-side navigation pane, choose **Instances > Instances**.
3. On the **Instances** page, click **Create Instance**.
   - **Organization**: Select an organization for the SLB instance from the drop-down list.

     > **Note** Make sure that the organization of the SLB instance is the same as the organization of its backend servers.

   - **Resource Set**: Select a resource set for the SLB instance from the drop-down list.
   - **Region**: Select the region where you want to deploy the SLB instance.
   - **Zone**: Select a zone for the SLB instance from the drop-down list.
   - **Instance Name**: Enter a name for the SLB instance in the Instance Name field.

The name must be 2 to 128 characters in length, and can contain letters, digits, full-width characters, hyphens (-), colons (:), periods (.), and underscores (_). Line breaks and spaces are supported. It must start with a letter and cannot start with `http://` or `https://`.

- **Instance Edition**: Select one of the following options: shared-performance and guaranteed-performance. Shared-performance SLB instances share resources with each other. The performance of shared-performance SLB instances is not guaranteed. The performance of guaranteed-performance SLB instances varies by type.
- **Instance Type**: Select the type of network traffic that you want to distribute. Valid values: Internal Network and Internet.
- **Network Type**: Select the network type of the SLB instance. Valid values: Classic Network and VPC.
- **IP Version**: Select an IP version.
- **IP Address**: Enter a service IP address for the SLB instance. Make sure that the service IP address is valid. Otherwise, the SLB instance cannot be created. If you do not set this parameter, the system automatically allocates an IP address to the SLB instance.

  > **Note** The private IP address that you specify must belong to the destination CIDR block of the vSwitch.

4. Click **Submit**.

What's next

Configure a CLB instance

20.1.4.3. Start and stop an SLB instance

This topic describes how to start and stop an SLB instance. SLB instances can be started or stopped at any time. A stopped SLB instance does not receive or forward client traffic.

Procedure

1. Log on to the SLB console.
2. In the left-side navigation pane, choose **Instances > Server Load Balancers**.
3. Find the target SLB instance. In the **Actions** column, choose  **> Start** or  **> Stop**.
4. To start or stop multiple instances at a time, select the instances and click **Start** or **Stop** at the bottom of the page.

20.1.4.4. Tags

20.1.4.4.1. Tag overview

This topic provides an overview of tags in SLB. SLB provides the tag management feature that allows you to classify SLB instances by using tags.

Each tag consists of a key and a value. Before you use tags, note the following limits:

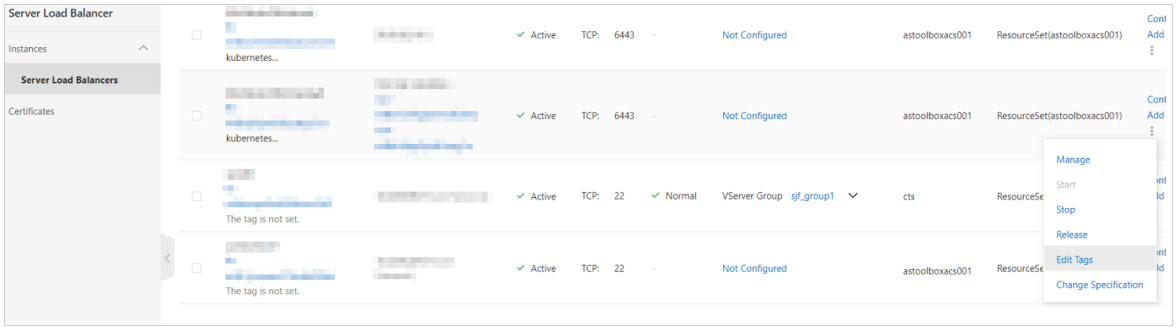
- Tags must be added to SLB instances.
- Each SLB instance can have a maximum of ten tags. You can add or remove a maximum of 5 tags at a time.
- The key of each tag added to an SLB instance must be unique. If a tag with the same key already exists, the tag is overwritten with the new value.

20.1.4.4.2. Add tags

This topic describes how to add tags to an SLB instance.

Procedure

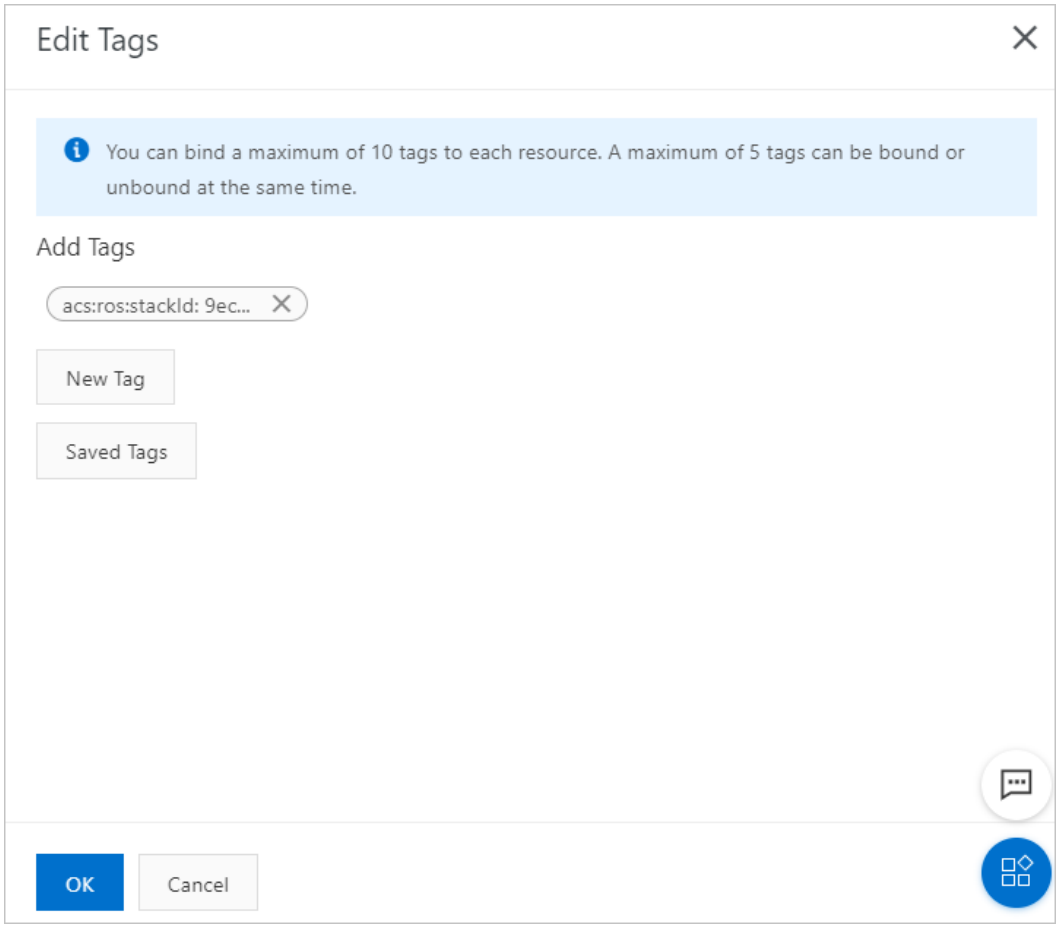
1. Log on to the SLB console.
2. In the left-side navigation pane, choose **Instances > Server Load Balancers**.
3. In the **Actions** column, choose > **Edit Tags**.



4. Edit tags in the **Edit Tags** dialog box.

   To add a tag, perform the following operations:

   - To add an existing tag, click **Saved Tags** and then select a tag.
   - To create and add a new tag, click **New Tag** in the **Edit Tags** dialog box, enter the key and value of the new tag, and then click **OK** next to the value.



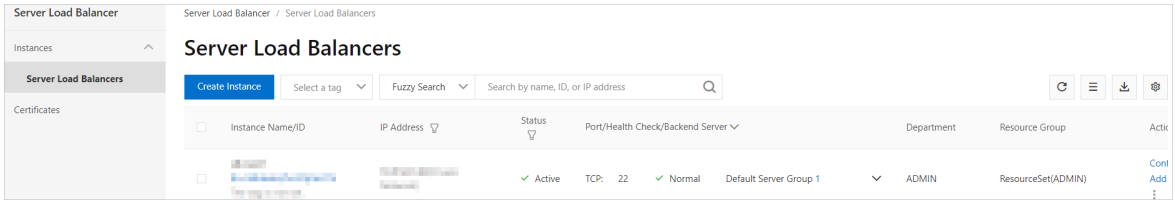
5. Click **OK**.

20.1.4.4.3. Query SLB instances by tag

This topic describes how to use tags to query SLB instances.

Procedure

1. Log on to the SLB console.
2. In the left-side navigation pane, choose **Instances > Server Load Balancers**.
3. Select a data from the **Select a tag** drip-down list to filter instances.



> **Note** To clear the search condition, move the pointer over the selected tag and click the displayed deletion icon next to it.

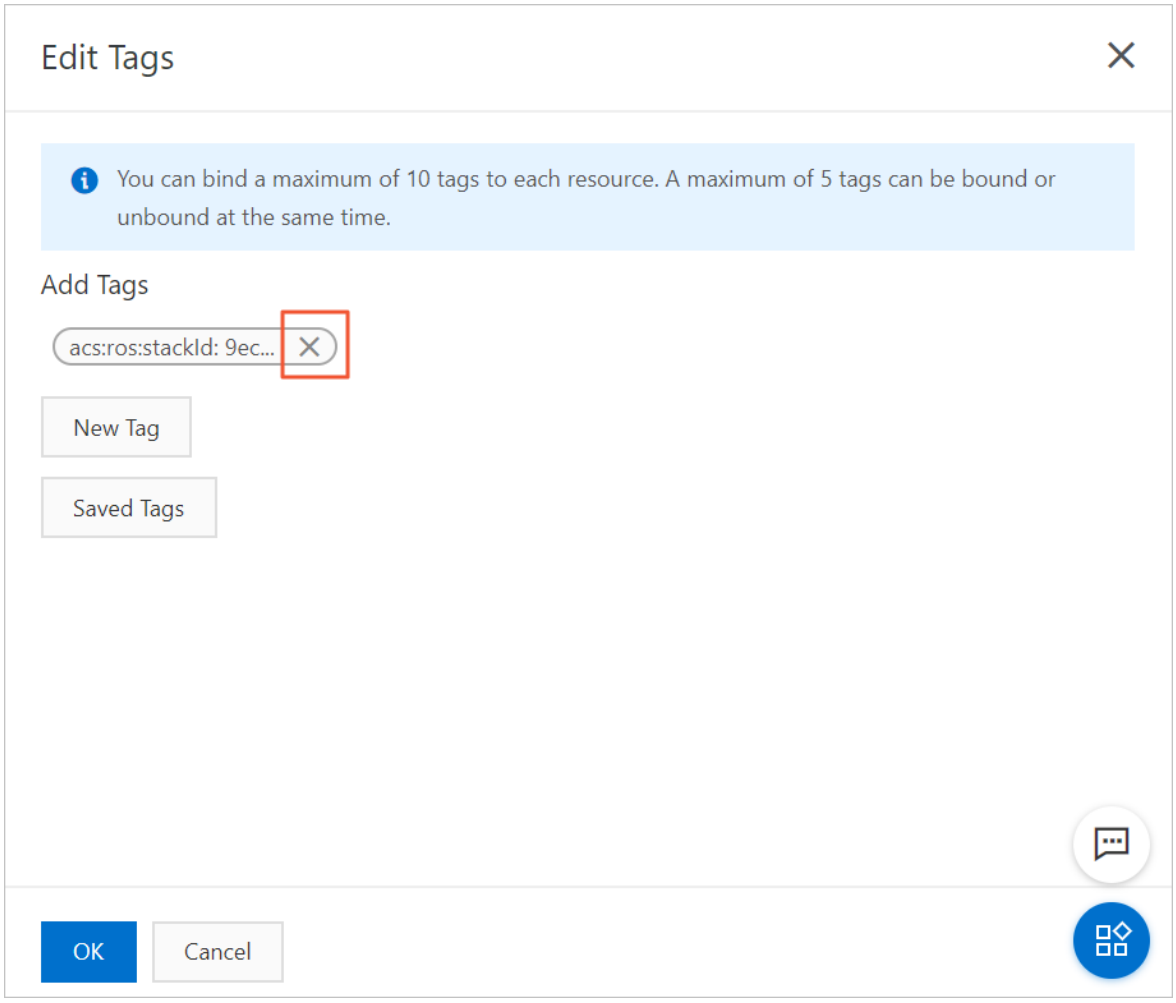
20.1.4.4.4. Remove tags

This topic describes how to remove tags from an SLB instance. You can only remove tags for one SLB instance at a time.

Procedure

1. Log on to the SLB console.
2. In the left-side navigation pane, choose **Instances > Server Load Balancers**.
3. In the **Actions** column, choose > **Edit Tags**.
4. In the **Edit Tags** dialog box, click the deletion icon next to the tags to be removed, and then click **OK**.

> **Note** If a tag is removed from an SLB instance and is not added to any other instances, the tag is deleted from the system.



20.1.4.5. Release an SLB instance

This topic describes how to release a Server Load Balancer (SLB) instance. You can immediately release SLB instances .

Procedure

1. Log on to the SLB console.
2. In the left-side navigation pane, choose **Instances > Instances**.
3. Find the SLB instance that you want to release and choose ⋮ **> Release** in the Actions column.
4. In the **Release** panel, **click Release Now**.
5. Click **Next**.
6. Confirm the displayed information and click **OK** to release the instance.

20.1.5. Listeners

20.1.5.1. Listener overview

This topic provides an overview of listeners. After you create a Server Load Balancer (SLB) instance, you must configure one or more listeners for it. A listener checks for connection requests and then distributes the requests to backend servers based on the forwarding rules that are defined by a specified scheduling algorithm.

SLB provides Layer 4 (TCP and UDP) and Layer 7 (HTTP and HTTPS) listeners. The following table lists the features and use cases of these listeners.

Protocol	Feature	Use case
TCP	• A connection-oriented protocol. A logical connection must be established before data can be sent and received. • Source IP address-based session persistence. • Source IP addresses readable at the network layer. • Fast data transmission	• Applicable to scenarios that require high reliability and data accuracy but can tolerate low speeds, such as file transmission, sending or receiving emails, and remote logons. • Web applications that do not have special requirements. For more information, see Add a TCP listener.
UDP	• A connectionless protocol. UDP transmits data packets directly instead of making a three-way handshake with the other party before UDP sends data. UDP does not provide error recovery or data re-transmission. • Fast data transmission but relatively low reliability.	Applicable to scenarios where real-time transmission is more important than reliability, such as video chats and real-time financial market pushes. For more information, see Add a UDP listener.
HTTP	• An application-layer protocol that is used to package data. • Cookie-based session persistence. • Use X-Forward-For to obtain source IP addresses.	Applicable to scenarios that require data content to be identified, such as web applications and small mobile games. For more information, see Add an HTTP listener.
HTTPS	• Encrypted data transmission that prevents unauthorized access. • Centralized certificate management service. You can upload certificates to SLB. The decryption operations are directly completed on SLB.	Applicable to scenarios that require encrypted transmission. For more information, see Add an HTTPS listener.

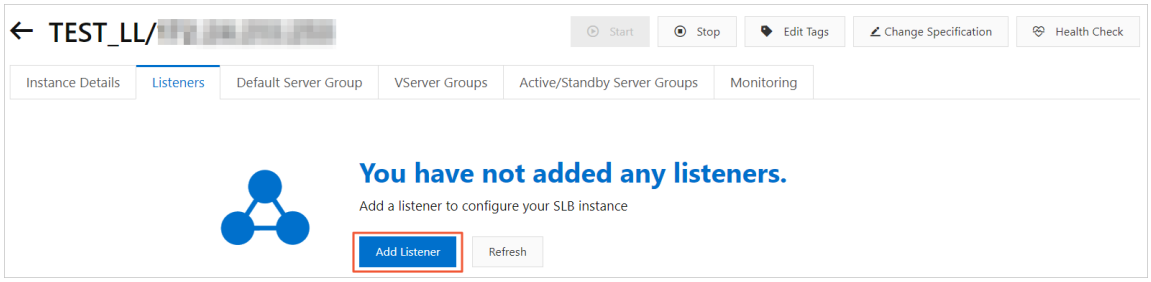
20.1.5.2. Add a TCP listener

This topic describes how to add a TCP listener to an SLB instance. TCP provides reliable and accurate data delivery at relatively low connection speeds and therefore is applicable to services such as file transmission, email sending or receiving, and remote logon. You can add a TCP listener to forward TCP requests.

Step 1: Start the listener configuration wizard

To start the listener configuration wizard, perform the following operations:

1. Log on to the SLB console.
2. In the left-side navigation pane, choose **Instances > Instances**.
3. Use one of the following methods to start the listener configuration wizard:
   - On the **Instances** page, find the SLB instance to which you want to add a TCP listener and then click **Configure Listener** in the Actions column.
   - On the **Instances** page, click the ID of the SLB instance. On the **Listener** tab, click **Add Listener**.



Step 2: Configure the TCP listener

To configure the TCP listener, perform the following operations:

1. Configure the following parameters.

Parameter	Description
Select Listener Protocol	Select the protocol of the listener. In this example, select **TCP**.
Listening Port	Set the listening port used to receive requests and forward them to backend servers. Valid values: 1 to 65535.
Advanced	
Scheduling Algorithm	SLB supports the following scheduling algorithms: RR, WRR, WLC. ○ **Weighted Round-Robin (WRR)**: Backend servers with a higher weight receive more requests than those with a lower weight. ○ **Round-Robin (RR)**: Requests are sequentially distributed to backend servers. ○ **Weighted Least Connections (WLC)**: Requests are distributed based on the combination of the weights and active connections of backend servers. If two backend servers have the same weight, requests are forwarded to the backend server with fewer connections.
Enable Session Persistence	Specify whether to enable session persistence. After session persistence is enabled, the listener forwards all requests from the same client to a specific backend server for the duration of a session. For TCP listeners, session persistence is implemented based on IP addresses. Requests from the same IP address are forwarded to the same backend server.
Enable Connection Draining	When connection draining is enabled, connections to the backend server that was removed or that is unhealthy are kept alive. **Note** The connection draining feature can be enabled only in the Australia (Sydney) and the UK (London) regions.

Parameter	Description
Connection Draining Timeout	After you enable connection draining, you can specify the maximum timeout period to keep connections alive before a backend server is removed from an SLB server group. After the backend server is removed or remains unhealthy for the specified period of time, SLB forcefully closes the connections to the server. Value values: 10 to 900. Unit: seconds.
Enable Peak Bandwidth Limit	Specify whether to set the bandwidth limit for the listener. If an SLB instance incurs fees based on the bandwidth, you can set different peak bandwidth values for different listeners to limit the amount of traffic that flows in each listener. The sum of the peak bandwidth values of all listeners added to an SLB instance cannot exceed the bandwidth of this SLB instance. By default, this feature is disabled and all listeners share the bandwidth of the SLB instance.
Idle Timeout	Specify the idle timeout period of TCP connections. Unit: seconds. Valid values: 10 to 900.
Obtain Client Source IP Address	For Layer 4 listeners, backend servers can directly obtain the actual IP addresses of clients.
Automatically Enable Listener After Creation	Specify whether to start the listener immediately after the listener is configured. By default, the listener is started after configuration.

2. Click **Next**.

Step 3: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can add backend servers to the default server group, or create VServer groups or primary/secondary server groups and then add servers to them. For more information, see Backend server overview.

The default server group is used in the example.

1. Select **Default Server Group** and click **Add More**.
2. On the **My Servers** panel, select ECS instances as the backend servers that you want to add and click **Next**.
3. In the **Configure Ports and Weights** step, configure weights for the added backend servers. A backend server that has a higher weight receives more requests.

   > **Note** If the weight of a backend server is set to 0, the backend server does not receive new requests.

4. Click **Add**. On the Default Server Group tab, configure ports for the backend servers. Valid values: 1 to 65535.

   You can specify the same port for multiple backend servers of an SLB instance.

5. Click **Next**.

Step 4: Configure health checks

SLB checks the availability of backend servers by performing health checks. The health check feature improves the overall availability of frontend services and avoid impacts on the service caused by exceptions of backend servers. Click **Modify** to configure advanced health check settings and click Next. For more information, see Health check overview.

Step 5: Submit the configurations

Perform the following operations to submit the configurations:

1. In the **Confirm** step, check the configurations. You can click **Modify** to modify the configurations.
2. Click **Submit**.
3. In the **Configuration Successful** message, click **OK**.

   You can check the created listener on the Listener tab.

20.1.5.3. Add a UDP listener

This topic describes how to add a UDP listener to an SLB instance. UDP is applicable to services that prioritize real-time content delivery over reliability, such as video chats and real-time quotes. You can add a UDP listener to forward UDP requests.

Context

Before you configure a UDP listener, take note of the following items:

- Ports 250, 4789, and 4790 of a UDP listener are reserved and therefore are unavailable for your configuration.
- Fragmented packets are not supported.
- The UDP listeners of an SLB instance in the classic network do not allow you to view source IP addresses.
- The following operations take five minutes to take effect if they are performed for a UDP listener:
  - Remove backend servers
  - Set the weight of a backend server to 0 after it is detected unhealthy

Step 1: Configure the UDP listener

To configure the UDP listener, perform the following operations:

1. In the **Protocol and Listener** step, configure the following parameters.

Parameter	Description
Select Listener Protocol	Select the protocol of the listener. In this example, select **UDP**.
Listening Port	Set the listening port that is used to receive requests and forward them to backend servers. Valid values: 1 to 65535.
Advanced	
Scheduling Algorithm	SLB supports three scheduling algorithms: RR and WRR. ○ **Weighted Round-Robin (WRR)**: Backend servers with a higher weight receive more requests than those with a lower weight. ○ **Round-Robin (RR)**: Requests are sequentially distributed to backend servers.

Parameter	Description
Enable Peak Bandwidth Limit	You can switch on this option and then set a bandwidth limit for the listener. If an SLB instance incurs fees based on the bandwidth, you can set different peak bandwidth values for different listeners to limit the amount of traffic that flows in each listener. The sum of the peak bandwidth values of all listeners added to an SLB instance cannot exceed the bandwidth of this SLB instance. By default, this feature is disabled and all listeners share the bandwidth of the SLB instance.
Obtain Client Source IP Address	Backend servers of a UDP listener can directly obtain the actual IP addresses of clients. **Note** UDP listeners of an SLB instance in the classic network do not allow you to view source IP addresses.
Automatically Enable Listener After Creation	Specify whether to start the listener after the listener is configured. By default, the listener is started after configuration.

2. Click **Next**.

Step 2: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can add backend servers to the default server group, or create VServer groups or primary/secondary server groups and then add servers to them. For more information, see Backend server overview.

Backend servers are added to the default server group in this example.

1. Select **Default Server Group** and click **Add More**.
2. Select ECS instances (backend servers) that you want to add, and then click **Next**.
3. Configure weights for the added backend servers.

   A backend server with a higher weight receives more requests.

   Note If the weight of a backend server is set to 0, the backend server does not receive new requests.

4. Click **Add**. On the **Default Server Group** tab, configure ports for the backend servers.

   Set a port for each backend server to receive requests. Valid values: 1 to 65535. You can specify the same port for multiple backend servers of an SLB instance.

5. Click **Next**.

Step 3: Configure health checks

CLB performs health checks to check the availability of the ECS instances that serve as backend servers. The health check feature improves overall service availability and reduces the impact of backend server failures.

On the **Health Check** wizard page, click **Modify** to modify the health check configurations. For more information, see Configure health checks.

Step 4: Submit the configurations

1. On the **Confirm** wizard page, check the configurations. You can click **Modify** to modify the configurations.
2. After you confirm the configurations, click **Submit**.
3. When Configuration Successful appears, click **OK**.

   After you configure the listener, you can view the listener on the Listener tab.

20.1.5.4. Add an HTTP listener

This topic describes how to add an HTTP listener to a Server Load Balancer (SLB) instance. HTTP is applicable to applications that need to identify data from different users, such as web applications and mini mobile games. You can add an HTTP listener to forward HTTP requests.

Step 2: Configure an HTTP listener

To configure an HTTP listener, perform the following operations:

1. On the **Protocol and Listener** wizard page, set the following parameters.

Parameter	Description
Select Listener Protocol	Select a protocol for the listener. **HTTP** is selected in this example.
Listening Port	Specify the listening port that is used to receive requests and forward them to backend servers. Valid values: 1 to 65535. **Note** You must not specify the same listening port for listeners that are associated with the same SLB instance.
Advanced	
Scheduling Algorithm	SLB supports three scheduling algorithms: round robin (RR) and weighted round robin (WRR). ○ **Weighted Round-Robin (WRR)**: Backend servers with higher weights receive more requests. ○ **Round-Robin (RR)**: Requests are sequentially distributed to backend servers.
Redirection	Specify whether to redirect traffic from the HTTP listener to an HTTPS listener. **Note** Before you enable redirection, make sure that you have created an HTTPS listener.

Parameter	Description
Enable Session Persistence	Specify whether to enable session persistence. After session persistence is enabled, SLB forwards all requests from a client to the same backend server. SLB maintains the persistence of HTTP sessions based on cookies. SLB allows you to use the following methods to process cookies: ◦ **Insert cookie**: You only need to specify the timeout period of the cookie. SLB inserts a cookie (SERVERID) into the first HTTP or HTTPS response packet that is sent to a client. The next request from the client contains this cookie and the listener distributes this request to the recorded backend server. ◦ **Rewrite cookie**: You can specify the cookie to be inserted into an HTTP or HTTPS response. You must specify the timeout period and lifecycle of this cookie on the backend server. After you specify a cookie, SLB overwrites the original cookie with the specified cookie. The next time SLB receives a client request that carries the specified cookie, the listener distributes the request to the recorded backend server.
Enable Peak Bandwidth Limit	Specify whether to enable bandwidth capping for the listener. If an SLB instance is billed based on bandwidth, you can set different maximum bandwidth values for different listeners to limit the amount of traffic that flows through each listener. The sum of the maximum bandwidth values of all listeners that are added to an SLB instance cannot exceed the bandwidth of the SLB instance. By default, this feature is disabled and all listeners share the bandwidth of the SLB instance.
Idle Timeout	Specify the timeout period of idle connections. Unit: seconds. Valid values: 1 to 60. If no request is received within the timeout period, SLB closes the connection. SLB recreates the connection after it receives a new connection request. This feature is available in all regions. **Note** This feature is unavailable for HTTP/2 requests.
Request Timeout	Specify the request timeout period. Unit: seconds. Valid values: 1 to 180. If no response is received from the backend server within the request timeout period, SLB returns an HTTP 504 error to the client. This feature is available in all regions.
Enable Gzip Compression	Specify whether to enable compression for a specified file type. Gzip supports the following file types: text/xml, text/plain, text/css, application/javascript, application/x-javascript, application/rss+xml, application/atom+xml, and application/xml.

Parameter	Description
Add HTTP Header Fields	You can add the following HTTP header fields: ○ Use the `X-Forwarded-For` header field to retrieve the IP addresses of clients. ○ Use the `X-Forwarded-Proto` header field to retrieve the listener protocol used by the SLB instance. ○ Use the `SLB-IP` header field to retrieve the public IP address of the SLB instance. ○ Use the `SLB-ID` header field to retrieve the ID of the SLB instance.
Obtain Client Source IP Address	HTTP listeners use the X-Forwarded-For header field to obtain the real IP addresses of clients.
Automatically Enable Listener After Creation	Specify whether to immediately enable the listener after the listener is created. By default, the listener is enabled after it is created.

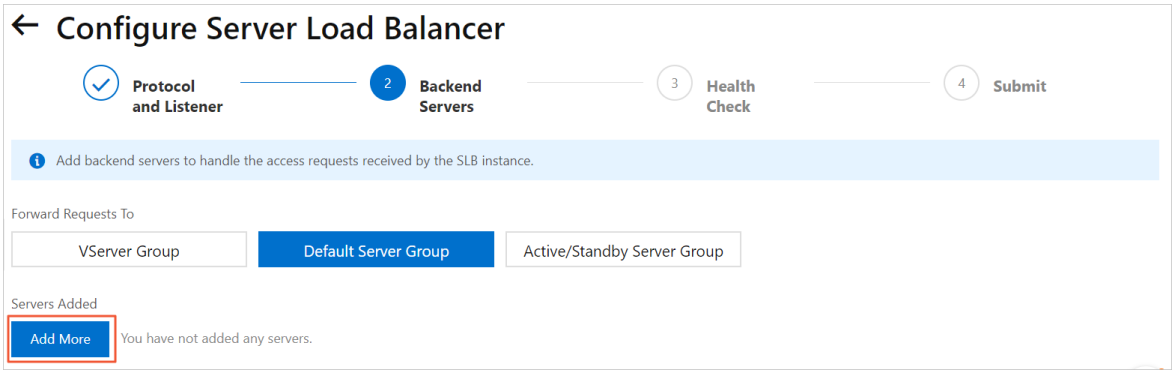
2. Click **Next**.

Step 3: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can use the default server group that is configured for the SLB instance, or create a vServer group or a primary/secondary server group. For more information, see Backend server overview.

Backend servers are added to the default server group in this example.

1. Select **Default Server Group** and click **Add More**.



2. In the **My Servers** panel, select the Elastic Compute Service (ECS) instances that you want to add as backend servers and click **Next**.
3. On the **Configure Ports and Weights** wizard page, specify the weights of the backend servers that you want to add. A backend server with a higher weight receives more requests.

> **Note** If the weight of a backend server is set to 0, no request is distributed to the backend server.

4. Click **Add**. On the Default Server Group tab, specify the ports that you want to open on the backend servers (ECS instances) to receive requests. Valid values: 1 to 65535.

   You can specify the same port on different backend servers that are connected to an SLB instance.
5. Click **Next**.

Step 3: Configure health checks

CLB performs health checks to check the availability of the ECS instances that serve as backend servers. The health check feature improves overall service availability and reduces the impact of backend server failures.

On the **Health Check** wizard page, click **Modify** to modify the health check configurations. For more information, see Configure health checks.

Step 4: Submit the configurations

1. On the **Confirm** wizard page, check the configurations. You can click **Modify** to modify the configurations.
2. After you confirm the configurations, click **Submit**.
3. When Configuration Successful appears, click **OK**.

   After you configure the listener, you can view the listener on the Listener tab.

20.1.5.5. Add an HTTPS listener

This topic describes how to add an HTTPS listener to a Server Load Balancer (SLB) instance. HTTPS is intended for applications that require encrypted data transmission. You can add an HTTPS listener to forward HTTPS requests.

Step 1: Configure a TCP listener

1.
2. Use one of the following methods to open the listener configuration wizard:
   - On the **Instances** page, find the CLB instance and click **Configure Listener** in the **Actions** column.
   - On the **Instances** page, find the CLB instance that you want to manage and click the ID of the instance. On the **Listener** tab, click **Add Listener**.
3. Set the following parameters and click **Next**.

Parameter	Description
Select Listener Protocol	Select **TCP**.
Listening Port	Set the listening port that is used to receive requests and forward them to backend servers. Valid values: 1 to 65535. You can set a TCP or UDP listener to listen on all ports within a specified port range.
Listener Name	Specify a name for the listener.
Advanced	Click **Modify** to configure advanced settings.

Parameter	Description
Scheduling Algorithm	Select a scheduling algorithm. ○ **Weighted Round-Robin (WRR)**: Backend servers that have higher weights receive more requests than backend servers that have lower weights. ○ **Round-Robin (RR)**: Requests are distributed to backend servers in sequence. ○ **Consistent Hash (CH)**: ■ **Tuple**: specifies consistent hashing that is based on four factors: source IP address, destination IP address, source port, and destination port. Requests that contain the same information based on the four factors are distributed to the same backend server. ■ **Source IP**: specifies consistent hashing that is based on source IP addresses. Requests from the same source IP address are distributed to the same backend server. **Note** Only high-performance CLB instances support the CH algorithm.
Enable Session Persistence	Specify whether to enable session persistence. After session persistence is enabled, CLB forwards all requests from a client to the same backend server. For TCP listeners, session persistence is implemented based on IP addresses. Requests from the same IP address are forwarded to the same backend server.
Enable Peak Bandwidth Limit	Specify whether to set the bandwidth limit of the listener. If a CLB instance is billed based on bandwidth usage, you can set different maximum bandwidth values for different listeners. This limits the amount of traffic that flows through each listener. The sum of the maximum bandwidth values of all listeners that are added to a CLB instance cannot exceed the maximum bandwidth value of the CLB instance. By default, this feature is disabled and all listeners share the bandwidth of the CLB instance.
Idle Timeout	Specify the timeout period of idle TCP connections. Unit: seconds. Valid values: 10 to 900.
Proxy Protocol	Use the proxy protocol to pass client IP addresses to backend servers. **Note** You cannot enable this feature in scenarios where PrivateLink is used.
Obtain Client Source IP Address	Specify whether to retrieve the real IP addresses of clients. Only Layer 4 listeners support this feature. By default, this feature is enabled.
Automatically Enable Listener After Creation	Specify whether to immediately enable the listener after it is created. By default, listeners are enabled after they are created.

Step 2: Configure an HTTPS listener

To configure an HTTPS listener, perform the following operations:

1. On the **Protocol and Listener** wizard page, set the following parameters.

Parameter	Description
Select Listener Protocol	Select a protocol for the listener. **HTTPS** is selected in this example.
Listening Port	Specify the listening port that is used to receive requests and forward them to backend servers. Valid values: 1 to 65535. **Note** You must not specify the same listening port for listeners that are associated with the same SLB instance.
Advanced	
Scheduling Algorithm	SLB supports three scheduling algorithms: round robin (RR), weighted round robin (WRR), and weighted least connections (WLC). ○ **Weighted Round-Robin (WRR)**: Backend servers with higher weights receive more requests. ○ **Round-Robin (RR)**: Requests are sequentially distributed to backend servers. ○ **Weighted Least Connections (WLC)**: Requests are distributed based on the weights and active connections of backend servers. Requests are distributed to the backend server with the least number of active connections. If two backend servers have the same number of active connections, the backend server with a higher weight receives more requests.
Enable Session Persistence	Specify whether to enable session persistence. After session persistence is enabled, SLB forwards all requests from a client to the same backend server. The timeout period for session persistence is 1 to 86,400 seconds. SLB maintains the persistence of HTTP sessions based on cookies. SLB allows you to use the following methods to process cookies: ○ **Insert cookie**: If you select this option, you only need to specify the timeout period of the cookie. SLB inserts a cookie (SERVERID) into the first HTTP or HTTPS response packet that is sent to a client. The next request from the client contains this cookie and the listener distributes this request to the recorded backend server. ○ **Rewrite cookie**: If you select this option, you can specify the cookie to be inserted into an HTTP or HTTPS response. You must specify the timeout period and lifecycle of the cookie on the backend server. After you specify a cookie, SLB overwrites the original cookie with the specified cookie. The next time SLB receives a client request that carries the specified cookie, the listener distributes the request to the recorded backend server.
Enable HTTP/2	Specify whether to enable HTTP/2.

Parameter	Description
Enable Peak Bandwidth Limit	Specify whether to enable bandwidth capping for the listener. If an SLB instance is billed based on bandwidth, you can set different maximum bandwidth values for different listeners to limit the amount of traffic that flows through each listener. The sum of the maximum bandwidth values of all listeners that are added to an SLB instance cannot exceed the bandwidth of this SLB instance. By default, this feature is disabled and all listeners share the bandwidth of the SLB instance.
Idle Timeout	Specify the timeout period of idle connections. Unit: seconds. Valid values: 1 to 60. If no request is received within the timeout period, SLB closes the connection. SLB recreates the connection after it receives a new connection request. This feature is available in all regions.
Request Timeout	Specify the request timeout period. Unit: seconds. Valid values: 1 to 180. If no response is received from the backend server within the request timeout period, SLB returns an HTTP 504 error to the client. This feature is available in all regions.
Enable Gzip Compression	Specify whether to enable Gzip compression for a specified file type. Gzip supports the following file types: text/xml, text/plain, text/css, application/javascript, application/x-javascript, application/rss+xml, application/atom+xml, and application/xml.
Add HTTP Header Fields	You can add the following HTTP header fields: ◦ Use the `X-Forwarded-For` header field to retrieve the IP addresses of clients. ◦ Use the `X-Forwarded-Proto` header field to retrieve the listener protocol used by the SLB instance. ◦ Use the `SLB-IP` header field to retrieve the public IP address of the SLB instance. ◦ Use the `SLB-ID` header field to retrieve the ID of the SLB instance.
Obtain Client Source IP Address	HTTP listeners use the X-Forwarded-For header field to retrieve the IP addresses of clients.
Automatically Enable Listener After Creation	Specify whether to immediately enable the listener after the listener is created. By default, the listener is enabled after it is created.

2. Click **Next**.

Step 3: Configure an SSL certificate

When you add an HTTPS listener, you must upload a server certificate or CA certificate, as shown in the following table.

Notice SLB supports the following public key algorithms:

- RSA 1024
- RSA 2048
- RSA 4096
- ECDSA P-256
- ECDSA P-384
- ECDSA P-521

Certificate	Description	Required for one-way authentication	Required for mutual authentication
Server certificate	The certificate that is used to identify the server. Your browser uses the server certificate to verify whether the certificate sent by the server is signed and issued by a trusted certification authority (CA).	Yes. You must upload the server certificate to the certificate management system of SLB.	Yes. You must upload the server certificate to the certificate management system of SLB.
Client certificate	The certificate that is used to identify the client. The server identifies the client by checking the certificate sent by the client. You can sign a client certificate with a self-signed CA certificate.	No.	Yes. You must install the client certificate on the client.
CA certificate	The server uses a CA certificate to verify the signature on the client certificate. If the signature is invalid, the connection request is denied.	No.	Yes. You must upload the CA certificate to the certificate management system of SLB.

Before you upload a certificate, take note of the following items:

- The certificate that you want to upload must be in the PEM format.
- After you upload a certificate to SLB, SLB can manage the certificate and you do not need to bind the certificate to backend servers.
- Time is needed to upload, load, and verify the certificate. Therefore, an HTTPS listener is not enabled immediately after it is created. It takes one to three minutes to enable an HTTPS listener.
- The ECDHE cipher suite used by HTTPS listeners supports forward secrecy. It does not support the security enhancement parameters that are required by the DHE cipher suite. Therefore, you cannot upload certificates (PEM files) that contain the `BEGIN DH PARAMETERS` field. For more information, see Certificate requirements.
- HTTPS listeners do not support Server Name Indication (SNI). You can choose TCP listeners and configure SNI on backend servers.
- The session ticket timeout period of HTTPS listeners is 300 seconds.
- The actual amount of data transfer on an HTTPS listener is larger than the billed amount because a portion of data is used for handshaking.
- The amount of data used for handshaking is large when a great number of connections are established.

The following section describes how to configure an SSL certificate. The international standard certificate and national standard certificate are taken as examples.

1. On the **SSL Certificates** wizard page, click **Switch Certificate Type** next to **International Standard Certificate Configuration**, select **International and National Standard Certificates**, and then click **OK**.
2. Select an uploaded server certificate from the **Select Server Certificate** drop-down list, a national standard encryption certificate from the **Select Server Encryption Certificate (National Standard Certificate)** drop-down list, and a national standard signature certificate from the **Select Server Signature Certificate (National Standard Certificate)** drop-down list. You can also click **Create Server Certificate** to upload a server certificate, national standard encryption certificate, and signature certificate.

   For more information, see Upload certificates.
3. To enable HTTPS mutual authentication, click **Modify** next to **Advanced**.
4. Turn on **Enable Mutual Authentication** and select an uploaded CA certificate, or click **Create CA Certificate** and upload a CA certificate.

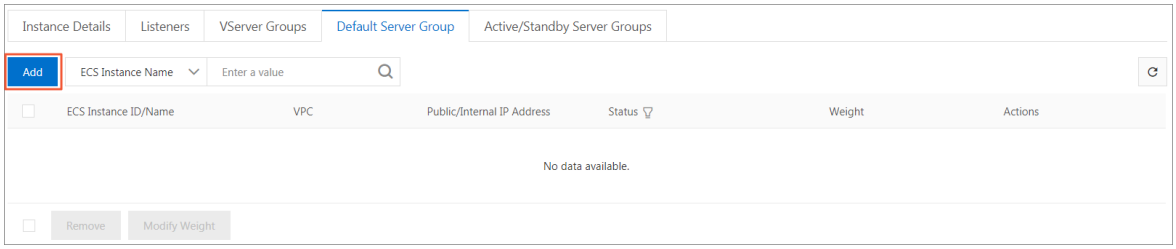
   You can use a self-signed CA certificate. For more information, see Overview.

Step 4: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can use the default server group that is configured for the SLB instance, or create a vServer group or a primary/secondary server group. For more information, see Backend server overview.

Backend servers are added to the default server group in this example.

1. Log on to the SLB console.
2. Find the SLB instance and click its instance ID.
3. Click the **Default Server Group** tab.
4. Click **Add**.



5. In the **My Servers** panel, select the Elastic Compute Service (ECS) instances that you want to add.
6. Click **Next**.
7. On the **Configure Ports and Weights** wizard page, specify the weight of each ECS instance.

   An ECS instance with a higher weight receives more requests.

   You can change the weight of a server and then move the pointer over  to change the weights of other servers:

   - Click **Replicate to Below**: The weights of all servers below the current server are set to the weight of the current server.
   - Click **Replicate to Above**: The weights of all servers above the current server are set to the weight of the current server.
   - Click **Replicate to All**: The weights of all servers in the default server group are set to the weight of the current server.
   - Click **Reset**: When you clear the weight of the current server, the weights of all servers in the default server group are cleared.

   > **Notice** After you set the weight of a server to 0, no request is distributed to the server.

8. Click **Add**.
9. Click **OK**.

Step 5: Configure health checks

CLB performs health checks to check the availability of the ECS instances that serve as backend servers. The health check feature improves overall service availability and reduces the impact of backend server failures.

Step 6: Confirm the configurations

To submit the configurations, perform the following operations:

1. On the **Confirm** wizard page, check the configurations. You can click **Modify** to modify the configurations.
2. After you confirm the configurations, click **Submit**.
3. When Configuration Successful appears, click **OK**.

   After you configure the listener, you can view the listener on the Listener tab.

20.1.5.6. Configure forwarding rules

This topic describes how to configure forwarding rules for a Server Load Balancer (SLB) instance. You can configure domain name-based or URL-based forwarding rules for an SLB instance that uses Layer 7 listeners. Layer 7 listeners distribute requests destined for different domain names or URLs to different Elastic Compute Service (ECS) instances.

Context

You can add multiple forwarding rules for a listener. Each forwarding rule is associated with a unique server group. Each server group contains multiple ECS instances. For example, you can configure a listener to forward read requests to one server group and write requests to another server group. This allows you to optimize load balancing among your server resources.

SLB forwards requests based on the following rules:

- If a request matches a domain name-based or URL-based forwarding rule of a listener, the request is forward to the corresponding server group based on the forwarding rule.
- If a request does not match any domain name-based or URL-based forwarding rules of a listener but the listener is associated with a server group, the request is forwarded to the server group.
- If none of the preceding conditions are met, requests are forwarded to backend ECS instances of the SLB instance based on the listener configuration.

Procedure

1. Log on to the SLB console.
2. Find the SLB instance that you want to manage and click the instance ID.
3. On the page that appears, click the **Listener** tab.
4. On the **Listeners** tab, find the listener that you want to manage.

   You can configure domain name-based or URL-based forwarding rules only for HTTP and HTTPS listeners.
5. In the **Actions** column, click the icon and select **Set Forwarding Rule** from the shortcut menu.
6. In the **Add Forwarding Rules** panel, click **Add Forwarding Rules**.
7. Configure forwarding rules based on the following information:
   - Configure a domain name-based forwarding rule
     - When you configure a domain name-based forwarding rule, leave the URL field empty. You do not need to enter a forward slash (/) in this field. The domain name can contain only letters, digits, hyphens (-), and periods (.).

- Domain-based forwarding rules support both exact match and wildcard match. For example, www.aliyun.com is an exact domain name while *.aliyun.com and *.market.aliyun.com are wildcard domain names. When a request matches multiple domain name-based forwarding rules, an exact match prevails over wildcard matches, as described in the following table. Domain name matching rule

Mode	Request URL	Domain name matching rule (√ indicates that the domain name is matched whereas x indicates that the domain name is not matched.)		
		www.aliyun.com	*.aliyun.com	*.market.aliyun.com
Exact match	www.aliyun.com	√	x	x
Wildcard match	market.aliyun.com	x	x	x
	info.market.aliyun.com	x	x	√

- Configure a URL-based forwarding rule
  - When you configure a URL-based forwarding rule, leave the Domain Name field empty.
  - The URL can contain only letters, digits, and hyphens `(-). /%? #&`
  - The URL must start with a forward slash (/).

    > **Note** If you enter only a forward slash (/) in the URL field, the URL-based forwarding rule is invalid.

  - URL-based forwarding rules support string matching and adopt sequential matching. Examples: `/admin`, `/bbs_`, and `/ino_test`.
- Configure both domain name-based and URL-based forwarding rules

  You can configure both domain name-based and URL-based forwarding rules when you need to forward traffic destined for different URLs of the same domain name. We recommend that you configure a default forwarding rule with the URL field left empty in case errors are returned when the URLs of requests are not matched.

  For example, the domain name of a website is `www.aaa.com`. You are required to forward requests destined for `www.aaa.com/index.html` to server group 1 and forward requests destined for other URLs of the domain name to server group 2. To meet the preceding requirements, you must configure two forwarding rules, as shown in the following figure. Otherwise, a 404 error code is returned when a request destined for the `www.aaa.com` domain name does not match any forwarding rules.

8. Click **Save**.

20.1.5.7. Enable access control

This topic describes how to enable access control for a listener. SLB provides listener-based access control. You can configure different whitelists for different listeners.

Procedure

1. Log on to the SLB console.
2. Find an SLB instance and click its instance ID.
3. On the page that appears, click the **Listener** tab.

4. Find the listener for which you want to enable access control, and choose > Set

Access Control in the Actions column.

5. In the **Access Control Settings** panel, turn on Enable Access Control and specify an ACL as a whitelist, and then click **OK**.

   Whitelist: Only the requests from the IP addresses or CIDR blocks in the specified ACL are forwarded. You can use the whitelist feature when you want to allow access from specified IP addresses.

   Risks may arise if you specify an ACL as a whitelist. After a whitelist is enabled, only IP addresses contained in the whitelist can access the SLB listener. If a whitelist is enabled without any IP addresses specified, the SLB listener forwards all requests.

   Separate multiple IP addresses with commas (,). Each IP address must be unique. You can add a maximum of 300 IP addresses.

   - IP addresses such as 10.23.12.24 and CIDR blocks such as 10.23.12.0/24 are supported.
   - 0.0.0.0 and x.x.x.x/0 are not supported.

   > **Note** The access control feature works only for new connection requests and does not affect existing connections.

20.1.5.8. Disable access control

This topic describes how to disable access control for a listener.

Procedure

1. Log on to the SLB console.
2. Find an SLB instance and click its instance ID.
3. Click the **Listener** tab next to the **Instance Details** tab.
4. Find the listener for which you want to disable access control, and choose > **Set Access Control** in the Actions column.
5. In the **Access Control Settings** panel, disable access control and then click **OK**.

20.1.6. Backend servers

20.1.6.1. Backend server overview

Before you use the Server Load Balancer (SLB) service, you must add Elastic Compute Service (ECS) instances as backend servers to an SLB instance to process client requests.

Backend server overview

You can set virtual IP addresses for an SLB instance. This way, the added ECS instances in the same region can be virtualized into an application service pool that provides high performance and availability. You can manage backend servers by using vServer groups. A listener of an SLB instance can be associated with a specific vServer group so that different listeners can forward requests to their associated backend servers that use different ports.

> **Note** If you associate a vServer group with a listener, the listener distributes requests to backend servers in the associated vServer group instead of those in the default server group.

Limits

You can increase or decrease the number of backend ECS instances at any time and switch ECS instances to receive client requests. When you perform the operations, make sure that the health check feature is enabled and at least one ECS instance is running as expected to maintain service stability.

When you add backend ECS instances, take note of the following items:

- You can add ECS instances of different operating systems to an SLB instance. However, the applications deployed on the ECS instances must be the same and have consistent data. We recommend that you use ECS instances of the same operating system to facilitate management and maintenance.
- Up to 50 listeners can be added to a single SLB instance. Each listener corresponds to an application deployed on backend ECS instances. Listening ports of an SLB instance correspond to application service ports that are opened on backend ECS instances.
- You can specify a weight for each ECS instance in the application service pool. An ECS instance with a higher weight receives more requests.
- If session persistence is enabled, requests may not be evenly distributed to backend ECS instances. To solve this problem, we recommend that you disable session persistence and check whether the problem persists.

  If requests are not evenly distributed, troubleshoot the issue in the following way:

  i. Collect statistics on the access logs of the web service on backend ECS instances for a specified period.
  ii. Check whether the numbers of access logs of backend ECS instances match SLB configurations. If session persistence is enabled, you must differentiate the access logs for the same IP address. If different weights are configured for backend ECS instances, you must check whether the percentage of access logs is normal based on the percentage of weights.
- When an ECS instance is undergoing hot migration, persistent connections to SLB may be interrupted. You can solve this problem by reestablishing the connections.

Default server groups

A default server group contains ECS instances that are used to receive requests. If a listener is not associated with a vServer group or a primary/secondary server group, the listener forwards requests to ECS instances in the default server group.

Before you use the SLB service, you must add at least one default backend server to receive client requests forwarded by SLB. For more information, see Add a default backend server.

vServer groups

You can use a vServer group if you want to distribute different requests to different backend servers or configure forwarding rules based on domain names and URLs. For more information, see Create a vServer group.

Primary/secondary server groups

A primary/secondary server group contains only two ECS instances. One ECS instance acts as the primary server and the other acts as the secondary server. Health checks are not performed on the secondary server. If the primary server is detected unhealthy, traffic is redirected to the secondary server. After the primary server recovers and is considered healthy, traffic is switched back to the primary server. For more information, see Create a primary/secondary server group.

> **Note** You can add primary/secondary server groups only for TCP and UDP listeners.

Related information

- Add a default backend server
- Create a vServer group
- Create a primary/secondary server group

20.1.6.2. Default server groups

20.1.6.2.1. Add ECS instances to the default server group

This topic describes how to add ECS instances as default backend servers to the default server group of an SLB instance. Before you use the SLB service, you must add at least one default backend server to receive client requests forwarded by SLB.

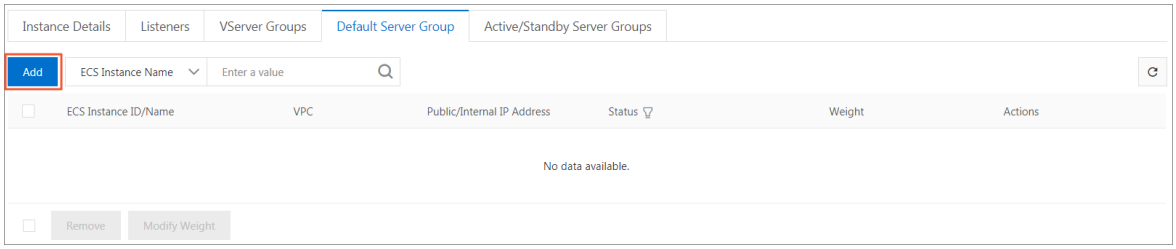
Prerequisites

Before you add ECS instances to the default server group, make sure that the following conditions are met:

- An SLB instance is created. For more information, see Create an SLB instance.
- You have created ECS instances and deployed applications on these ECS instances to process requests.

Procedure

1. Log on to the SLB console.
2. Find the target SLB instance and click its instance ID.
3. Click the **Default Server Group** tab.
4. Click **Add**.



5. In the **My Servers** dialog box, select ECS instances that you want to add.
6. Click **Next**.
7. In the **Configure Ports and Weights** dialog box, specify the weight of each added ECS instance.

   An ECS instance with a higher weight receives more requests.

   You can change the weight of a server and then move the pointer over to change the weights of multiple servers:

   - Click **Replicate to Below**: The weights of all servers below the current server are set to the weight of the current server.
   - Click **Replicate to Above**: The weights of all servers above the current server are set to the weight of the current server.
   - Click **Replicate to All**: The weights of all servers in the default server group are set to the weight of the current server.
   - Click **Reset**: The weight fields of all servers in the default server group are cleared.

> **Notice** If the weight of a backend server is set to 0, this backend server no longer receives new requests.

8. Click **Add**.
9. Click **OK**.

20.1.6.2.2. Add IDC servers to the default server group

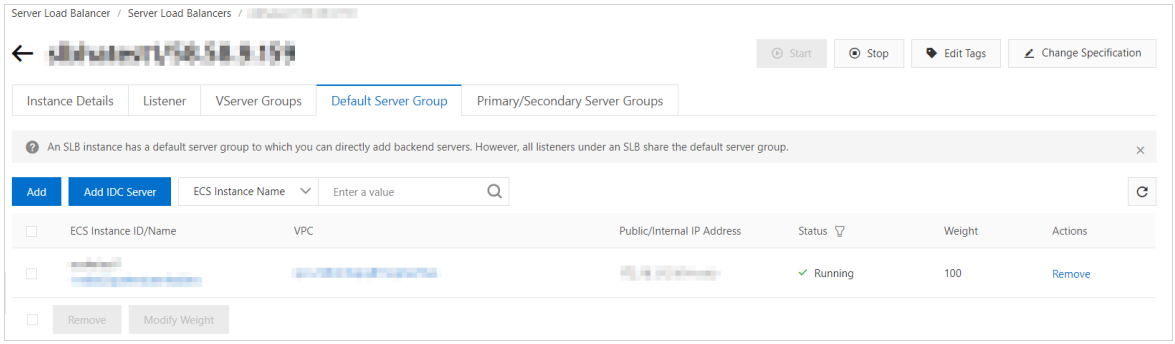
This topic describes how to add servers in on-premises Internet Data Centers (IDCs) as default backend servers to the default server group of an SLB instance. Before you use the SLB service, you must add at least one default backend server to receive client requests forwarded by SLB.

Prerequisites

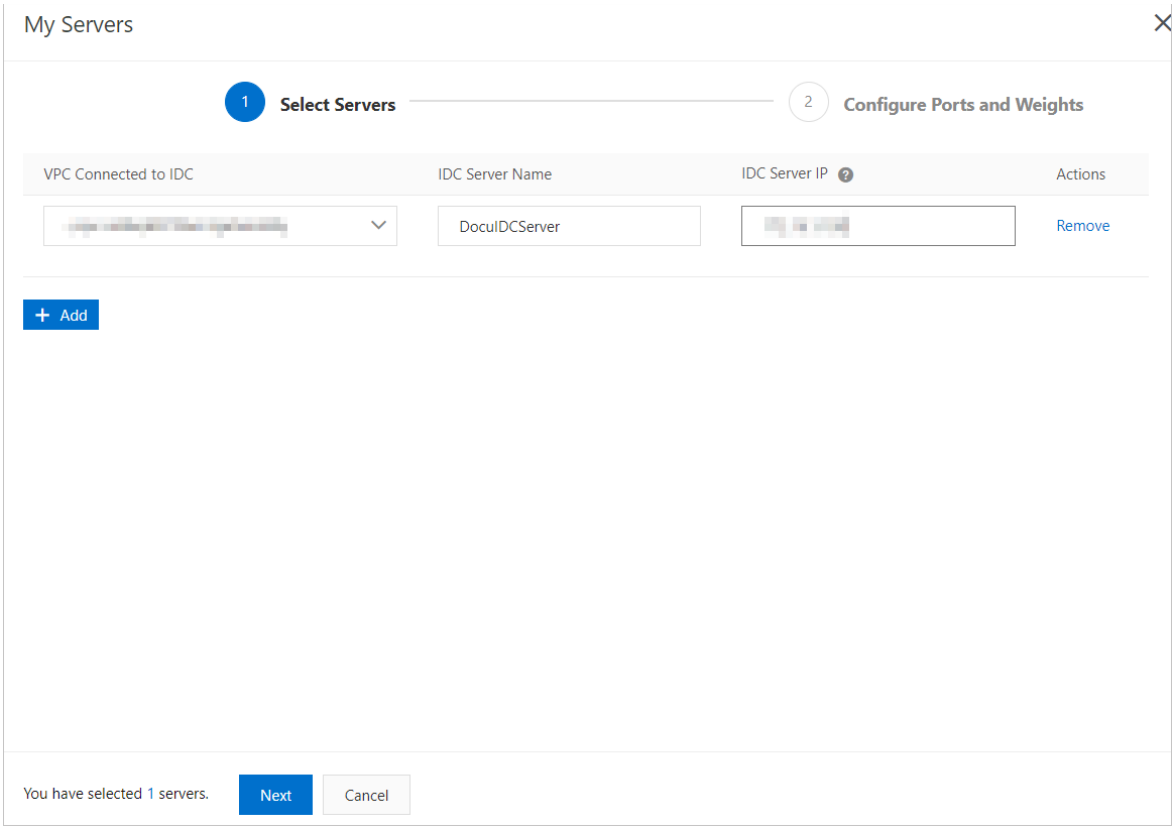
Applications are deployed on the IDC servers, and the IDC servers are ready to receive distributed requests.

Procedure

1. Log on to the SLB console.
2. Find the target SLB instance and click its instance ID.
3. Click the **Default Server Group** tab.
4. Click **Add IDC Server**.



5. In the **My Servers** dialog box, click **Add**.
6. Select a VPC from the VPC Connected to IDC drop-down list, enter a name for the IDC server, and specify the IP address of the IDC server.



7. Click **Next**.
8. In the **Configure Ports and Weights** step, specify the weight of each added IDC server.

   An IDC server with a higher weight receives more requests.

   You can change the weight of a server and then move the pointer over to change the weights of multiple servers:

   - Click **Replicate to Below**: The weights of all servers below the current server are set to the weight of the current server.
   - Click **Replicate to Above**: The weights of all servers above the current server are set to the weight of the current server.
   - Click **Replicate to All**: The weights of all servers in the default server group are set to the weight of the current server.
   - Click **Reset**: The weight fields of all servers in the default server group are cleared.

   > **Notice** If the weight of a backend server is set to 0, this backend server no longer receives new requests.

9. Click **Add**.
10. Click **OK**.

20.1.6.2.3. Change the weight of a backend server

This topic describes how to change the weight of a backend server to adjust the proportion of requests sent to the backend server.

Procedure

1. Log on to the SLB console.
2. Find the target SLB instance and click its instance ID.
3. Click the **Default Server Group** tab.
4. Move the pointer over the weight value of the target backend server and click the  icon.
5. Change the weight and then click **OK**.

   A backend server (ECS instance or IDC server) with a higher weight receives more requests.

   > **Notice** The weight value ranges from 0 to 100. If the weight of a backend server is set to 0, no requests are sent to the backend server.

20.1.6.2.4. Remove a backend server

This topic describes how to remove a backend server that is no longer needed.

Procedure

1. Log on to the SLB console.
2. Find the target SLB instance and click its instance ID.
3. Click the **Default Server Group** tab.
4. Find the target backend server and click **Remove** in the **Actions** column.
5. In the dialog box that appears, click **OK**.

20.1.6.3. VServer groups

20.1.6.3.1. Create a vServer group

This topic describes how to create a vServer group for a Server Load Balancer (SLB) instance. A vServer group contains Elastic Compute Service (ECS) instances that function as backend servers. If you associate a vServer group with a listener, the listener distributes requests only to backend servers in the vServer group.

Prerequisites

Before you create a vServer group, make sure that the following conditions are met:

- An SLB instance is created. For more information, see Create an SLB instance.
- ECS instances are created and applications are deployed on the ECS instances to process requests.

Context

Take note of the following items before you create a vServer group for an SLB instance:

- ECS instances are added to a vServer group and the corresponding SLB instance must be deployed in the same region.
- An ECS instance can be added to multiple vServer groups.
- A vServer group can be associated with multiple listeners of an SLB instance.
- A vServer group consists of ECS instances and application ports.

Procedure

1. Log on to the SLB console.
2. Find the SLB instance and click its instance ID.

3. Click the **VServer Groups** tab.
4. On the **VServer Groups** tab, click **Create VServer Group**.
5. On the **Create VServer Group** page, set the parameters.
   i. In the **VServer Group Name** field, enter a name for the vServer group.
   ii. Click **Add**. On the **My Servers** wizard page, select the ECS instances that you want to add.
   iii. Click **Next**.
   iv. Set the Port and Weight parameters for each ECS instance, and then click **Add**.

   Set the Port and Weight parameters based on the following information:

   - **Port**: The backend port opened on an ECS instance to receive requests.

     You can set the same port number for multiple backend servers of an SLB instance.
   - **Weight**: An ECS instance with a higher weight receives more requests.

     > **Notice** If the weight of an ECS instance is set to 0, the ECS instance no longer receives new requests.

   You can click  to specify the ports and weights of the added ECS instances in batches.

   - **Replicate to Below**: The ports or weights of all servers below the current server are set to the port or weight of the current server.
   - **Replicate to Above**: The ports or weights of all servers above the current server are set to the port or weight of the current server.
   - **Replicate to All**: The ports or weights of all servers in the vServer group are set to the port or weight of the current server.
   - **Reset**: If the port or weight of the current server is cleared, the ports or weights of all servers in the vServer group are also cleared.
6. Click **Create**.

20.1.6.3.2. Add IDC servers to a VServer group

This topic describes how to create a VServer group and then add IDC servers to the VServer group. You can add ECS instances and IDC servers as backend servers to a VServer group. If you associate a VServer group with a listener, the listener distributes requests only to the backend servers in the VServer group instead of other backend servers.

Prerequisites

Before you create a VServer group, make sure that applications are deployed on the IDC servers and the IDC servers are ready to receive distributed requests.

Context

Note the following items before you create a VServer group:

- An IDC server can be added to multiple VServer groups.
- A VServer group can be associated with multiple listeners of an SLB instance.
- The settings of the VServer group include the settings of IDC servers and application ports.

Procedure

1. Log on to the SLB console.
2. Find the target SLB instance and click its instance ID.

3. Click the **VServer Groups** tab.
4. On the **VServer Groups** tab, click **Create VServer Group**.
5. On the **Create VServer Group** page, configure the VServer group.
   i. In the **VServer Group Name** field, enter a name for the VServer group.
   ii. Click **Add IDC Server**.
   iii. In the **My Servers** dialog box, click **Add**.
   iv. Select a VPC from the VPC Connected to IDC drop-down list, enter a name for the IDC server, and specify the IP address of the IDC server.

   The IP address of the IDC server must be accessible to the VPC.
   v. Click **Next**.
   vi. Specify a port and weight for each IDC server, and then click **Add**.

   Set the ports and weights based on the following information:

   - **Port**: The backend port opened on an IDC server to receive requests. Multiple ports can be added to an IDC server.

     You can set the same port number for multiple backend servers of the same SLB instance.
   - **Weight**: An IDC server with a higher weight receives more requests.

     > **Notice** If the weight of an IDC server is set to 0, the IDC server no longer receives new requests.

   You can change the weight of a server and then move the pointer over  to change the weights of multiple servers:

   - Click **Replicate to Below**: The weights of all servers below the current server are set to the weight of the current server.
   - Click **Replicate to Above**: The weights of all servers above the current server are set to the weight of the current server.
   - Click **Replicate to All**: The weights of all servers in the VServer group are set to the weight of the current server.
   - Click **Reset**: The weight fields of all servers in the VServer group are cleared.

     > **Notice** If the weight of a backend server is set to 0, this backend server no longer receives new requests.

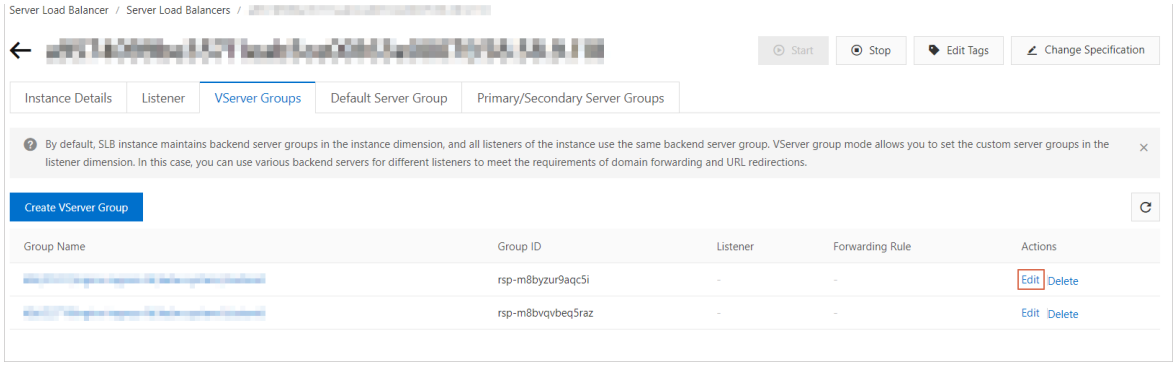
   vii. Click **Add**.
6. Click **Create**.

20.1.6.3.3. Modify a VServer group

This topic describes how to modify the settings of ECS instances or IDC servers in a VServer group.

Procedure

1. Log on to the SLB console.
2. Find the target SLB instance and click its instance ID.
3. Click the **VServer Groups** tab.
4. Find the target VServer group and then click **Edit** in the Actions column.



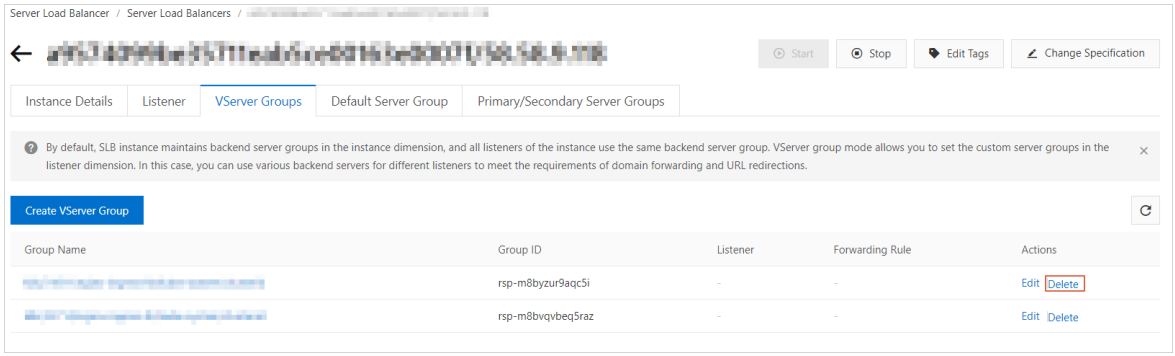
5. Modify the ports and weights of ECS instances or IDC servers, and then click **Save**.

20.1.6.3.4. Delete a VServer group

This topic describes how to delete a VServer group that is no longer needed for traffic distribution.

Procedure

1. Log on to the SLB console.
2. Find the target SLB instance and click its instance ID.
3. Click the **VServer Groups** tab.
4. Find the target VServer group, and then click **Delete** in the Actions column.



5. In the dialog box that appears, click **OK**.

20.1.6.4. Active/standby server groups

20.1.6.4.1. Create a primary/secondary server group

This topic describes how to create a primary/secondary server group and then add Elastic Compute Service (ECS) instances to the primary/secondary server group. A primary/secondary server group contains a primary server and a secondary server that can fail over to prevent service interruption. By default, the primary server handles all requests that are distributed by the SLB instance. When the primary server fails, requests are redirected to the secondary server.

Prerequisites

Before you create a primary/secondary server group, make sure that the following requirements are met:

- A Server Load Balancer (SLB) instance is created. For more information, see Create an SLB instance.
- ECS instances are created and applications are deployed on the ECS instances to process requests.

> **Note** Only TCP and UDP listeners support primary/secondary server groups.

Procedure

1. Log on to the SLB console.
2. Find the SLB instance and click its instance ID.
3. Click the **Primary/Secondary Server Groups** tab.
4. On the **Primary/Secondary Server Groups** tab, click **Create Primary/Secondary Server Group**.
5. On the **Create Primary/Secondary Server Group** page, configure the primary/secondary server group.
   i. In the **Primary/Secondary Server Group Name** field, enter a name for the primary/secondary server group.
   ii. Click **Add**. In the **My Servers** panel, select the ECS instances that you want to add.

   You can add only two ECS instances to a primary/secondary server group.
   iii. Click **Next**.
   iv. On the **Configure Ports and Weights** wizard page, specify the backend ports that you want to open on the ECS instances to receive requests. If you want to open more than one port on an ECS instance, click **Add Port** in the **Actions** column.

   You can open the same port on different backend servers that are connected to the same SLB instance.
   v. Click **Add**.
6. On the **Create Primary/Secondary Server Group** page, select an ECS instance in the **Type** column as the primary server.
7. Click **Create**.

20.1.6.4.2. Add IDC servers to a primary/secondary server group

This topic describes how to create a primary/secondary server group and then add IDC servers to the primary/secondary server group. You can use a primary/secondary server group to implement failover between a primary server and a secondary server. By default, the primary server handles all distributed requests. When the primary server fails, traffic is redirected to the secondary server.

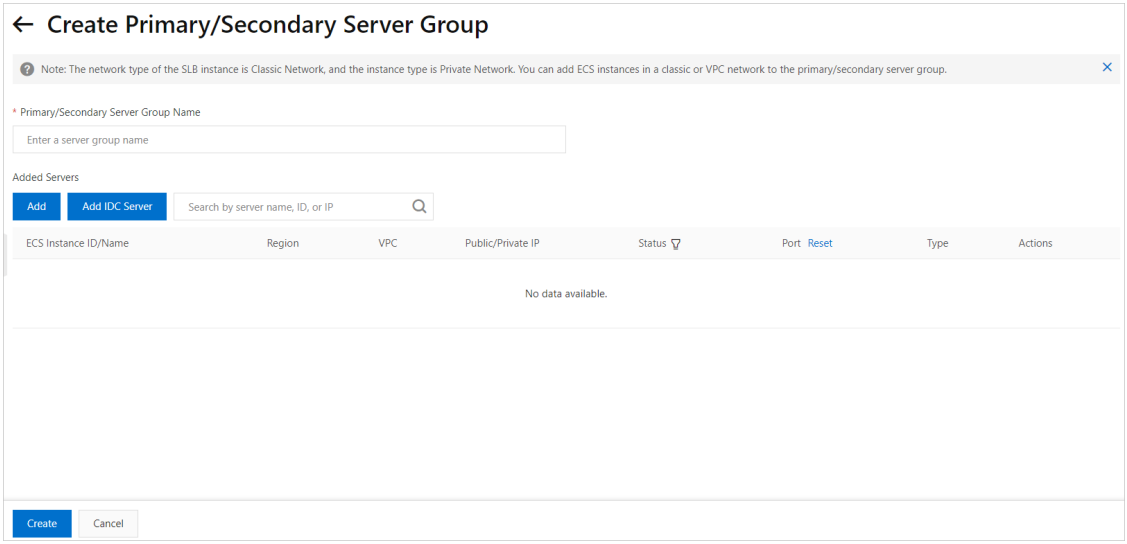
Prerequisites

The IDC servers are created, configured to deploy applications, and ready to receive distributed requests.

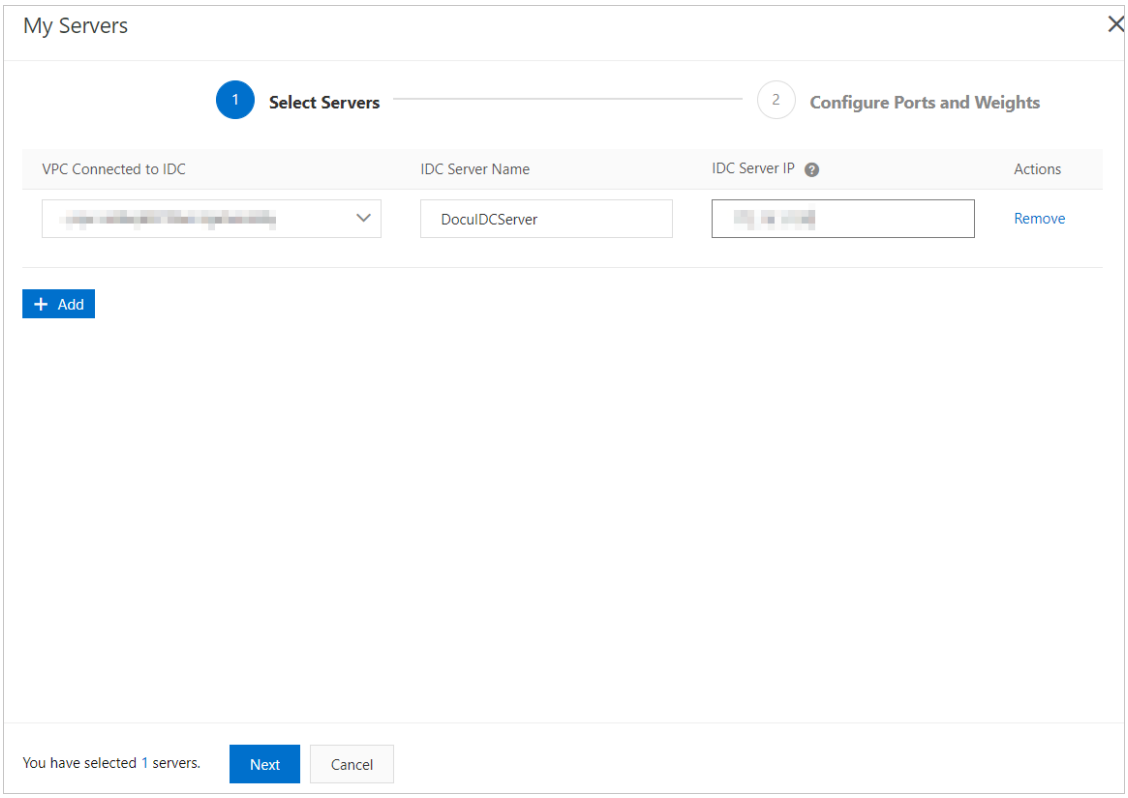
Procedure

1. Log on to the SLB console.
2. Find the target SLB instance and click its instance ID.
3. Click the **Primary/Secondary Server Groups** tab.
4. On the **Primary/Secondary Server Groups** tab, click **Create Primary/Secondary Server Group**.
5. On the **Create Primary/Secondary Server Group** page, configure the primary/secondary server group.

i. In the **Primary/Secondary Server Group Name** field, enter a name for the primary/secondary server group, and then click **Add IDC Server**.



ii. In the **My Servers** dialog box, click **Add**.

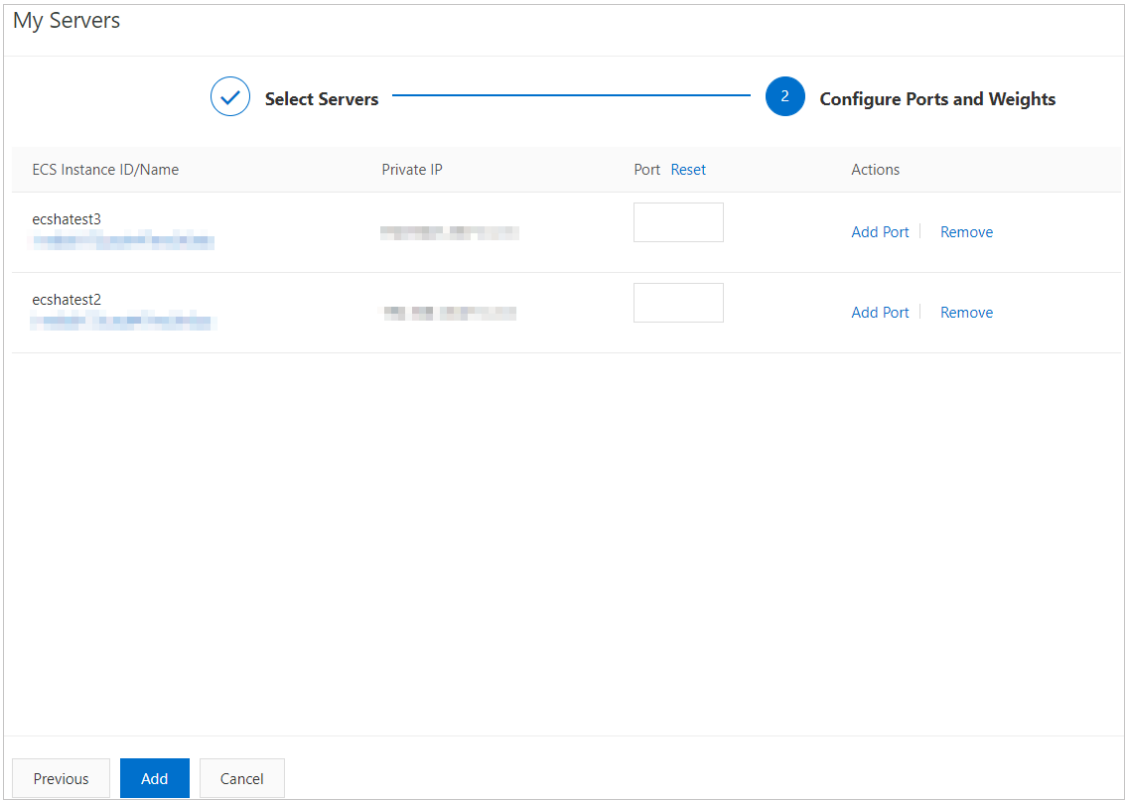


iii. Select a VPC from the VPC Connected to IDC drop-down list, enter a name for the IDC server, and specify the IP address of the IDC server.

The IP address of the IDC server must be accessible to the VPC.

iv. Click **Next**.

v. Configure the backend ports opened on ECS instances to receive requests, and then click **Add**.



You can set multiple ports for an IDC server.

vi. Set a backend server as the primary server.

vii. Click **Create**.

20.1.6.4.3. Delete a primary/secondary server group

This topic describes how to delete a primary/secondary server group of a Server Load Balancer (SLB) instance. If a primary/secondary server group is no longer needed to forward traffic, you can delete the primary/secondary server group.

Procedure

1. Log on to the SLB console.
2. Find the SLB instance and click its instance ID.
3. Click the **Primary/Secondary Server Groups** tab.
4. On the **Primary/Secondary Server Groups** tab, find the primary/secondary server group that you want to delete and click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

20.1.7. Health check

20.1.7.1. Health check overview

This topic describes the health check feature of Server Load Balancer (SLB). SLB checks the availability of Elastic Compute Service (ECS) instances that act as backend servers by performing health checks. The health check feature improves the overall availability of your frontend business and mitigates the impacts of exceptions that occur on backend ECS instances.

After you enable the health check feature, SLB stops distributing requests to ECS instances that are declared unhealthy and distributes new requests to healthy ECS instances. When the unhealthy ECS instances have recovered, SLB starts forwarding requests to these ECS instances again.

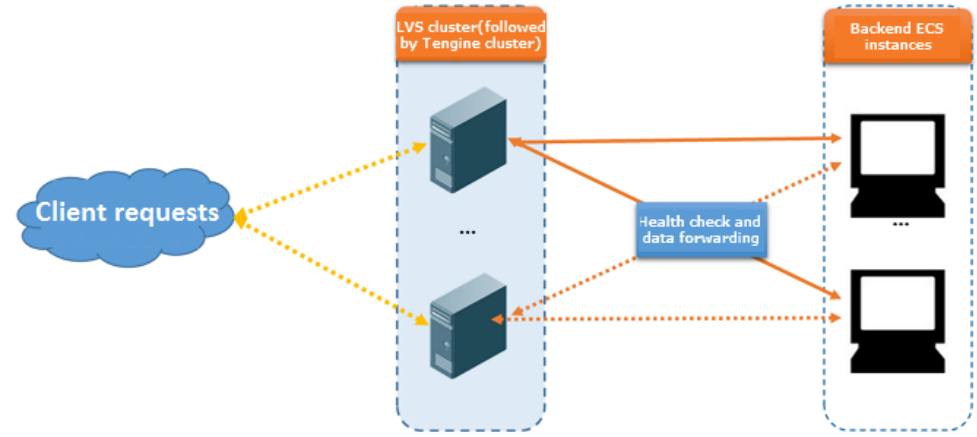
If your business is highly sensitive to traffic loads, frequent health checks may impact the availability of normal business. To reduce the impacts of health checks on your business, you can reduce the health check frequency, increase the health check interval, or change Layer 7 health checks to Layer 4 health checks. We recommend that you do not disable the health check feature to ensure business continuity.

Health check process

SLB is deployed in clusters. Node servers in the LVS or Tengine cluster forward data and perform health checks.

The node servers in the LVS cluster forward data and perform health checks independently and in parallel based on configured load balancing policies. If an LVS node server detects that a backend ECS instance is unhealthy, this node server no longer sends new client requests to this ECS instance. This operation is synchronized among all node servers in the LVS cluster.

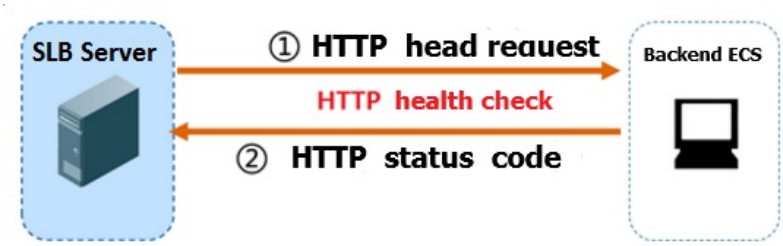
SLB uses the CIDR block of 100.64.0.0/10 for health checks. Make sure that backend ECS instances do not block this CIDR block. You do not need to configure a security group rule to allow access from this CIDR block. However, if you have configured security rules such as iptables, you must allow access from this CIDR block. 100.64.0.0/10 is reserved by Alibaba Cloud. Other users cannot use any IP addresses within this CIDR block, and therefore no relevant security risks exist.



Health checks of HTTP or HTTPS listeners

For Layer 7 (HTTP or HTTPS) listeners, SLB checks the status of backend ECS instances by sending HTTP HEAD requests. The following figure shows the process.

For HTTPS listeners, certificates are managed in SLB. To improve system performance, HTTPS is not used for data exchange (including health check data and business interaction data) between SLB and backend ECS instances.



The following section describes the health check process of a Layer 7 listener:

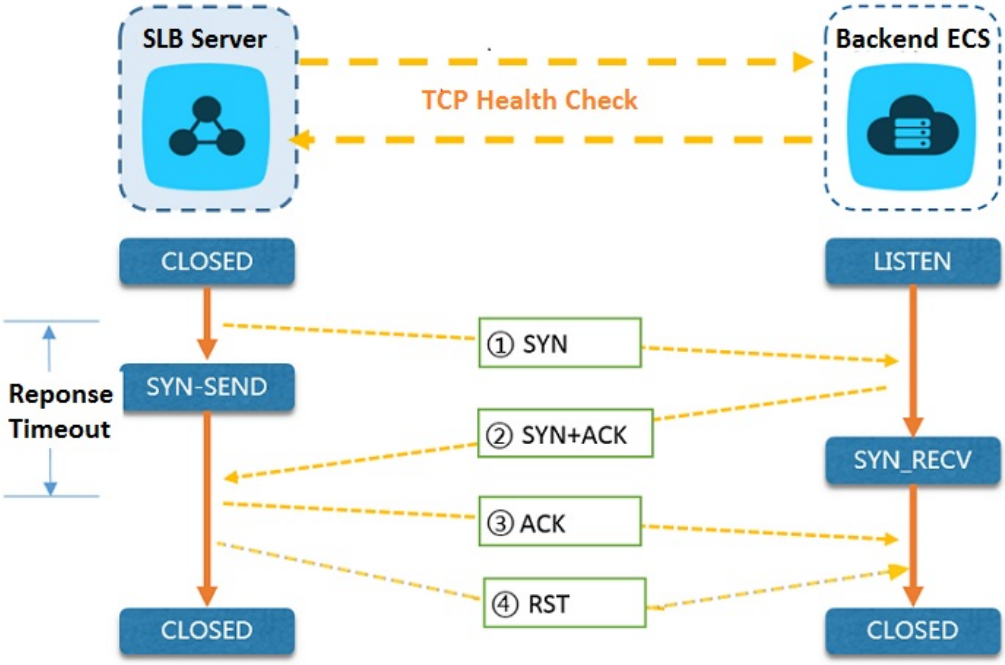
1. A Tengine node server sends an HTTP HEAD request that contains the configured domain name to the internal IP address, health check port, and health check path of a backend ECS instance based on health check settings.
2. After the backend ECS instance receives the request, the ECS instance returns an HTTP status code based on

the running status.

3. If the Tengine node server does not receive a response from the backend ECS instance within the specified response timeout period, the backend server is declared unhealthy.
4. If the Tengine node server receives a response from the backend ECS instance within the specified response timeout period, the node server compares the response with the configured status code. If the response contains the status code that indicates a healthy server, the backend server is declared healthy. Otherwise, the backend server is declared unhealthy.

Health checks of TCP listeners

For TCP listeners, SLB checks the status of backend servers by establishing TCP connections to improve health check efficiency. The following figure shows the process.



The following section describes the health check process of a TCP listener:

1. An LVS node server sends a TCP SYN packet to the internal IP address and health check port of a backend ECS instance.
2. After the backend ECS instance receives the request, the ECS instance returns an SYN-ACK packet if the corresponding port is listening normally.
3. If the LVS node server does not receive a packet from the backend ECS instance within the specified response timeout period, the backend ECS instance is declared unhealthy. Then, the node server sends an RST packet to the backend ECS instance to terminate the TCP connection.
4. If the LVS node server receives a packet from the backend ECS instance within the specified response timeout period, the node server determines that the service runs properly and the health check succeeds. Then, the node server sends an RST packet to the backend ECS instance to terminate the TCP connection.

Note A TCP three-way handshake is conducted to establish a TCP connection. After the LVS node server receives the SYN+ACK packet from the backend ECS instance, the node server sends an ACK packet, and then immediately sends an RST packet to terminate the TCP connection.

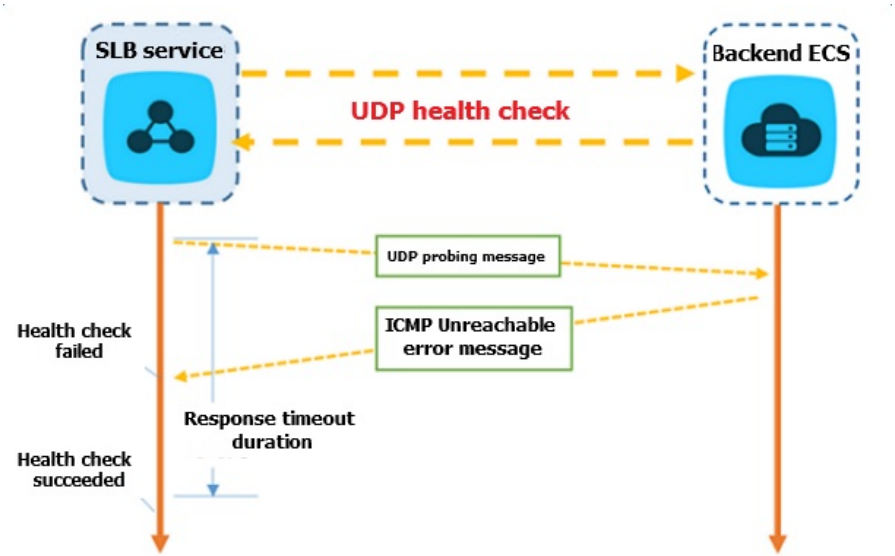
This process may cause backend ECS instances to think that an error such as an abnormal exit occurred in the TCP connection. Then, these instances may report a corresponding error message, such as `Connection reset by peer`, in logs such as Java connection pool logs.

Solution:

- You can implement HTTP health checks.
- If you have enabled the feature of obtaining actual client IP addresses on backend ECS instances, you can ignore connection errors caused by the access of the SLB CIDR block.

Health checks of UDP listeners

For UDP listeners, SLB checks the status of backend ECS instances by sending UDP packets. The following figure shows the process.



The following section describes the health check process of a UDP listener:

1. An LVS node server sends a UDP packet to the internal IP address and health check port of an ECS instance based on health check configurations.
2. If the corresponding port of the ECS instance is not listening normally, the system returns an ICMP error message, such as `port XX unreachable`. Otherwise, no message is returned.
3. If the LVS node server receives the ICMP error message within the response timeout period, the backend ECS instance is declared unhealthy.
4. If the LVS node server does not receive any messages from the backend ECS instance within the response timeout period, the ECS instance is declared healthy.

Note For UDP health checks, the health check result may not reflect the real status of a backend ECS instance in the following situation:

If the backend ECS instance uses a Linux operating system, the speed at which ICMP messages in high concurrency scenarios are sent is limited due to the ICMP attack prevention feature of Linux. In this case, even if a service exception occurs, SLB may declare the backend ECS instance healthy because the error message `port XX unreachable` is not returned. Consequently, the health check result deviates from the actual service status.

Solution:

You can specify a request and a response for UDP health checks. The ECS instance is considered healthy only when the specified response is returned. However, the client must be configured accordingly to return responses.

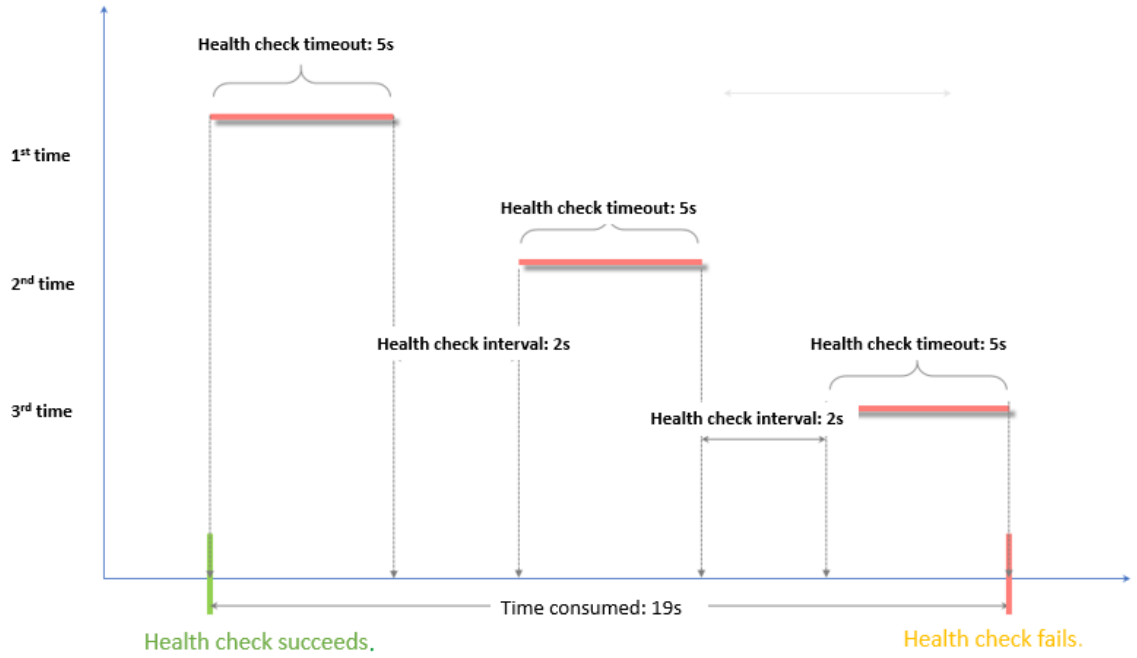
Health check time window

The health check feature effectively improves the availability of your services. However, to avoid impacts on system availability caused by frequent switching after failed health checks, the health check status switches only when health checks successively succeed or fail for a specified number of times within a certain time window. The health check time window is determined by the following factors:

- Health check interval: how often health checks are performed
- Response timeout: the length of time to wait for a response
- Health check threshold: the number of consecutive successes or failures of health checks

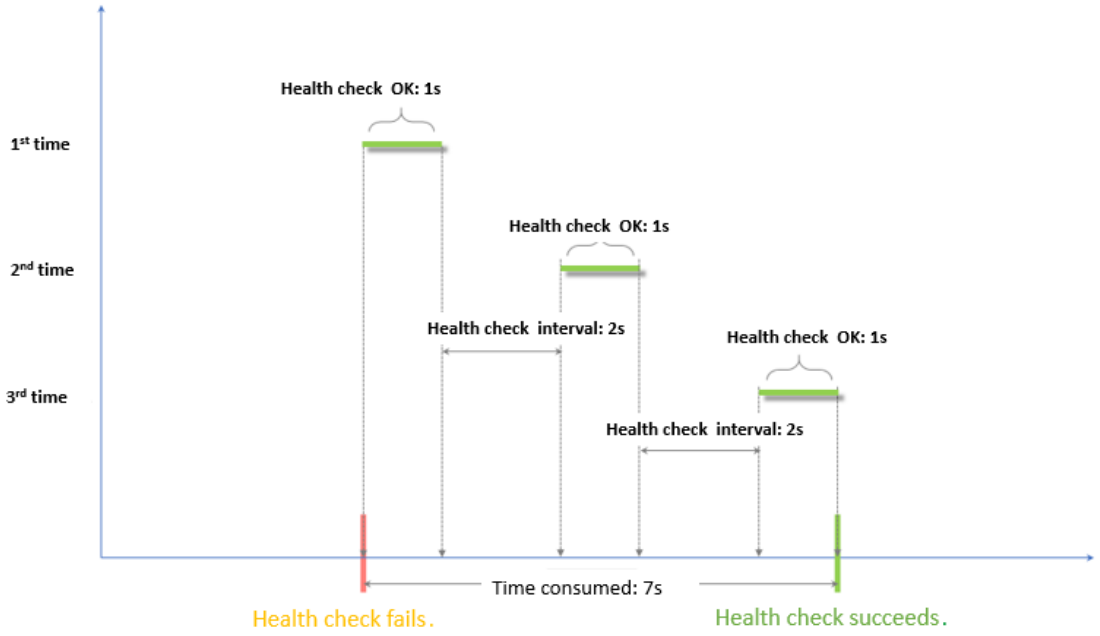
The health check time window is calculated based on the following formula:

- Time window for health check failures = Response timeout × Unhealthy threshold + Health check interval × (Unhealthy threshold - 1)



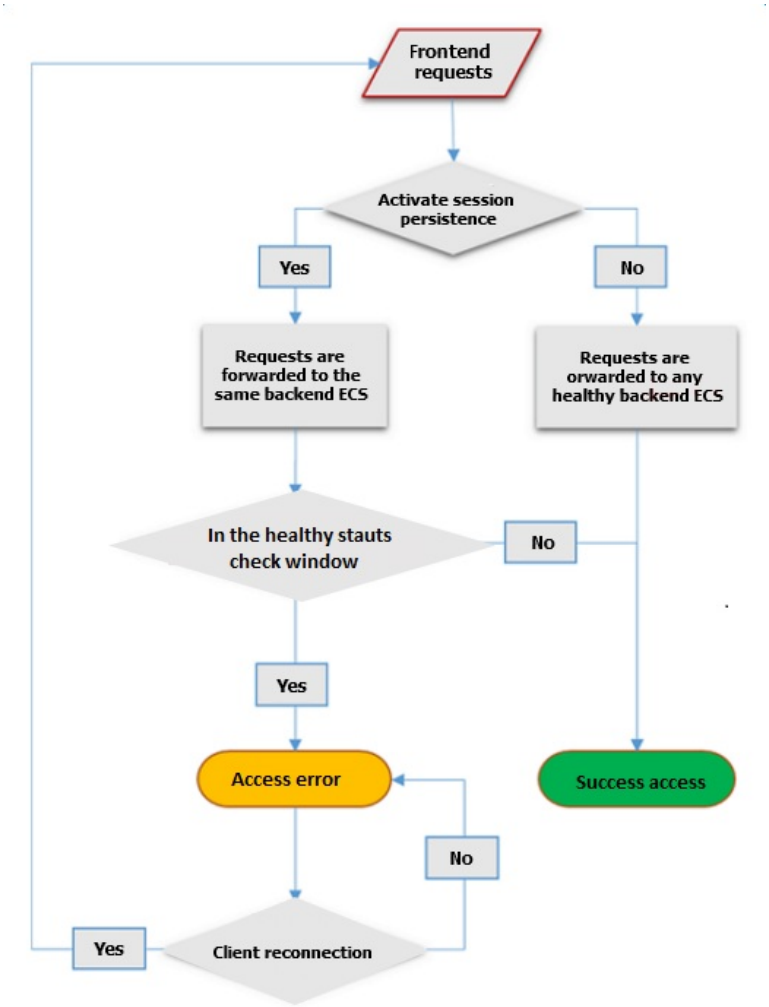
- Time window for health check successes = Response time of a successful health check × Healthy threshold + Health check interval × (Healthy threshold - 1)

> **Note** The response time of a successful health check is the duration from the time when the health check request is sent to the time when the response is received. When TCP health checks are used, the response time is short and almost negligible because only whether the specific port is alive is checked. For HTTP health checks, the response time depends on the performance and load of the application server and is typically within a few seconds.



The health check result has the following impacts on request forwarding:

- If the health check of the backend ECS instance fails, new requests are distributed to other backend ECS instances. This does not affect client access.
- If the health check of the backend ECS instance succeeds, new requests are distributed to this instance. The client access is normal.
- If an exception occurs on the backend ECS instance and a request arrives during a time window for health check failures, the request is still sent to the backend ECS instance. This is because the number of failed health checks has not reached the unhealthy threshold (3 times by default). In this case, the client access fails.



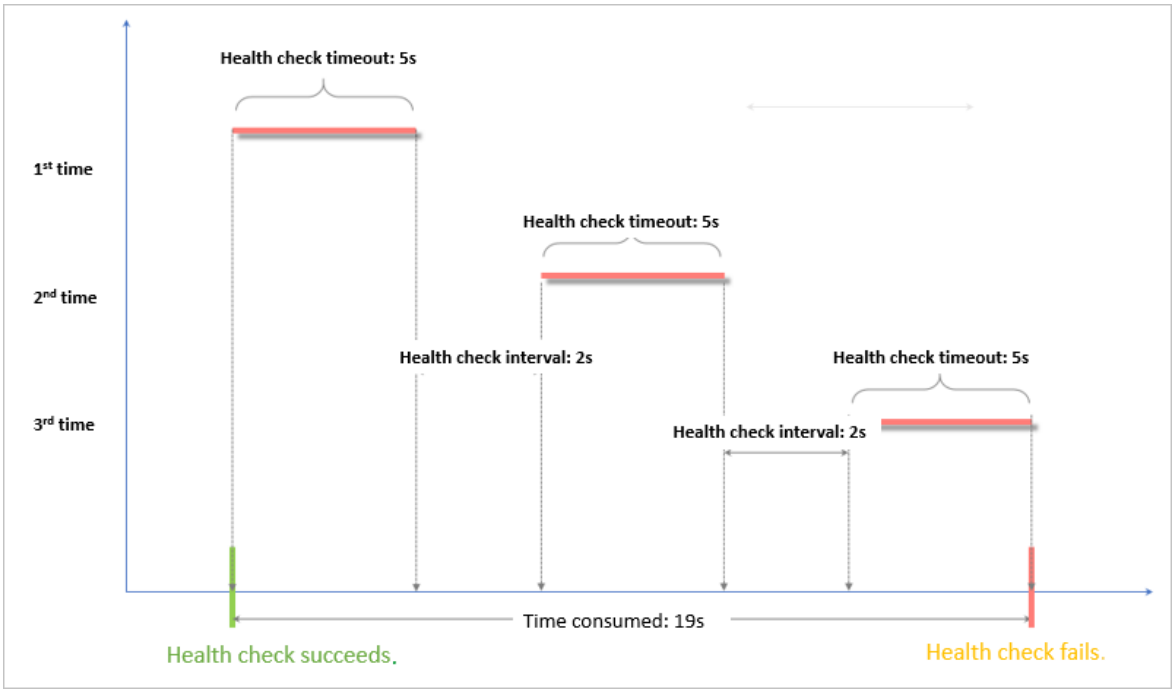
Examples of health check response timeout and health check interval

The following health check settings are used in these examples:

- **Response Timeout Period: 5 Seconds**
- **Health Check Interval: 2 Seconds**
- **Healthy Threshold: 3 Times**
- **Unhealthy Threshold: 3 Times**

Time window for health check failures = Response timeout × Unhealthy threshold + Health check interval × (Unhealthy threshold - 1). That is, 5 × 3 + 2 × (3 - 1) = 19 seconds. If the response time of a health check exceeds 19 seconds, the health check fails.

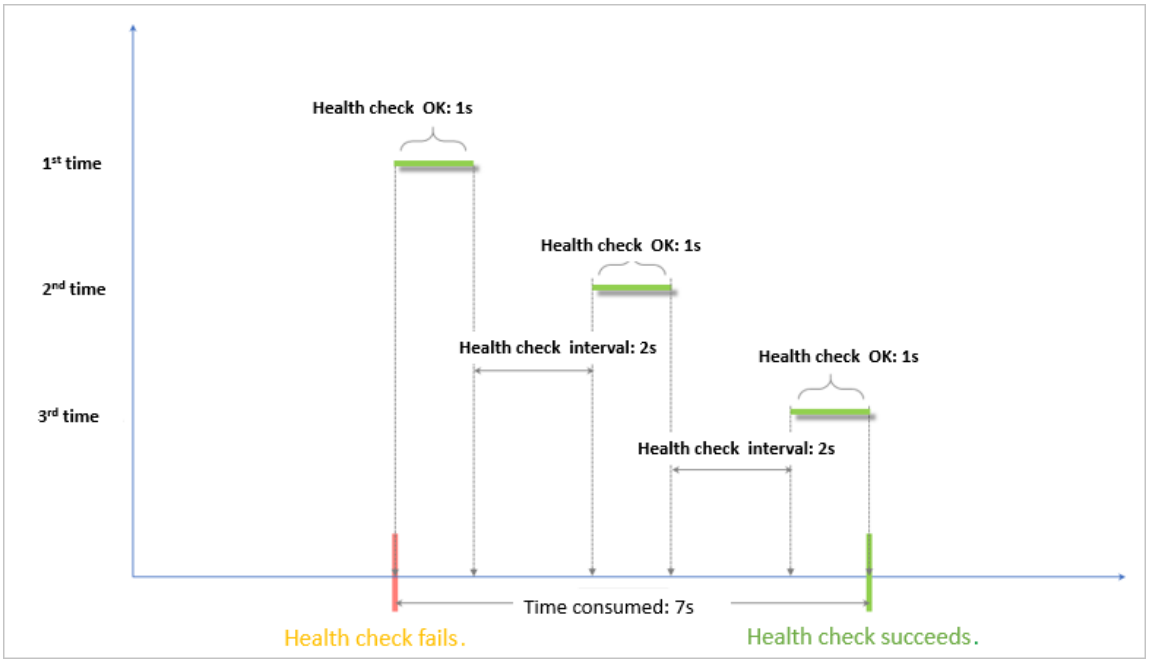
The following figure shows the time window from a healthy status to an unhealthy status.



Time window for health check successes = Response time of a successful health check × Healthy threshold + Health check interval × (Healthy threshold - 1). That is, (1 × 3) + 2 × (3 - 1) = 7 seconds. If the response time of a successful health check is less than seven seconds, the health check succeeds.

> **Note** The response time of a successful health check is the duration from the time when the health check request is sent to the time when the response is received. When TCP health checks are used, the response time is short and almost negligible because only whether the specific port is alive is checked. For HTTP health checks, the response time depends on the performance and load of the application server and is typically within a few seconds.

The following figure shows the time window from an unhealthy status to a healthy status (assume that it takes 1 second for the server to respond to a health check request).



Domain name setting in HTTP health checks

When HTTP health checks are used, you can set a domain name for health checks. The setting is optional. Some application servers verify the host field in requests. In this case, the request header must contain the host field. If a domain name is configured in health check setting, SLB adds the domain name to the host field when SLB forwards a request to an application server. If no domain name is configured, the health check request is denied by the application server because it does not contain a host field and the health check may fail. If your application server verifies the host field in requests, you must configure a domain name to make sure that the health check feature works.

20.1.7.2. Configure health checks

This topic describes how to configure health checks. You can configure health checks when you create a listener or for an existing listener. The default health check settings can meet your requirements in most cases.

Procedure

1. Log on to the SLB console.
2. Find an SLB instance and click the instance ID.
3. On the page that appears, click the **Listener** tab.
4. Click **Add Listener**, or find an existing listener and click **Modify Listener** in the **Actions** column.
5. Click **Next** to go to the **Health Check** step and configure the health check.

   We recommend that you use the default settings when you configure health checks.

   Health check parameters

Parameter	Description
Health Check Protocol	Select the protocol that the SLB instance uses when it performs health checks. For TCP listeners, both TCP health checks and HTTP health checks are supported. ○ A TCP health check implements detection at the network layer by sending SYN packets to check whether a port is open. ○ An HTTP health check verifies the health of a backend server by sending HEAD or GET requests to simulate browser access.
Health Check Method (for the HTTP and HTTPS health checks only)	Health checks of Layer 7 (HTTP or HTTPS) listeners support both the HEAD and GET methods. The HEAD method is used by default. If your backend application server does not support the HEAD method or if the HEAD method is disabled, the health check may fail. To solve this issue, you can use the GET method instead. If the GET method is used and the response size exceeds 8 KB, the response is truncated. However, the health check result is not affected.

Parameter	Description
Health Check Path and **Health Check Domain Name (Optional)** (for the HTTP health checks only)	By default, SLB sends HTTP HEAD requests to the default homepage configured on the application server through the internal IP address of the backend ECS instance to perform health checks. If you do not use the default homepage of the application server for health checks, you must specify the path for health checks. Some application servers verify the host field in requests. In this case, the request header must contain the host field. If a domain name is configured in health check settings, SLB adds this domain name to the host field when SLB forwards a health check request to one of the preceding application servers. If no domain name is configured, SLB does not include the host field in requests and the requests are rejected by the application server, which may cause health checks to fail. If your application server verifies the host field in requests, you must configure a domain name in health check settings to ensure that the health check feature functions properly.
Normal Status Code (for the HTTP health checks only)	Select the HTTP status code that indicates successful health checks. Default values: http_2xx and http_3xx.
Health Check Port	The detection port used by the health check feature to access backend servers. By default, the backend port configured for the listener is used. **Note** If a VServer group or a primary/secondary server group is configured for the listener, and the ECS instances in the group use different ports, leave this parameter empty. SLB uses the backend port of each ECS instance to perform health checks.
Response Timeout	The length of time to wait for a health check response. If the backend ECS instance does not send an expected response within the specified period of time, the health check fails. Valid values: 1 to 300. Unit: seconds. Default value for UDP listeners: 10. Default value for HTTP, HTTPS, and TCP listeners: 5.
Health Check Interval	The interval between two consecutive health checks. All nodes in the LVS cluster perform health checks independently and in parallel on backend ECS instances at the specified interval. The health check statistics of a single ECS instance cannot reflect the health check interval because the nodes perform health checks at different times. Valid values: 1 to 50. Unit: seconds. Default value for UDP listeners: 5. Default value for HTTP, HTTPS, and TCP listeners: 2.
Unhealthy Threshold	The number of consecutive failed health checks that must occur on a backend ECS instance before this ECS instance is declared unhealthy. Valid values: 2 to 10. Default value: 3.

6. Click **Next**.

20.1.7.3. Disable the health check feature

This topic describes how to disable the health check feature for a Server Load Balancer (SLB) instance. If you disable the health check feature, requests may be distributed to unhealthy backend Elastic Compute Service (ECS) instances. This causes service disruptions. We recommend that you enable the health check feature.

Procedure

1. Log on to the SLB console.
2. On the **Instances** page, find the SLB instance that you want to manage and click its instance ID.
3. On the **Listener** tab, find the listener for which you want to disable the health check feature and click **Modify Listener** in the **Actions** column.
4. On the **Configure Listener** page, click **Next** to proceed to the **Health Check** wizard page.
5. Turn off the Enable Health Check switch and click **Next**.
6. Click **Submit** and click **OK**.

20.1.8. Certificate management

20.1.8.1. Certificate overview

This topic provides an overview of the certificates that can be deployed on SLB instances. To use an HTTPS listener, you must upload the required third-party server certificate and digital identification issued by a certificate authority (CA) to SLB. You do not need to configure certificates on backend servers after uploading the certificates to SLB.

To upload a third-party certificate, you must have the files that contain the public key and private key of the certificate.

HTTPS server certificates and client CA certificates are supported.

You can create a maximum of 100 certificates per account.

20.1.8.2. Certificate requirements

Server Load Balancer (SLB) supports only certificates in the PEM format. Before you upload a certificate, make sure that the certificate content, certificate chain, and private key meet the corresponding format requirements.

Certificates issued by a root CA

If the certificate was issued by a root certification authority (CA), the received certificate is the only one that needs to be uploaded to SLB. In this case, the website that is configured with this certificate is regarded as a trusted website and does not require additional certificates.

The certificate must meet the following format requirements:

- The certificate must start with `-----BEGIN CERTIFICATE-----` and end with -----END CERTIFICATE-----.
- Each line (except the last line) must contain 64 characters. The last line can contain 64 or fewer characters.
- The certificate content cannot contain spaces.

Certificates issued by an intermediate CA

If the certificate was issued by an intermediate CA, the received certificate file contains multiple certificates. You must upload both the server certificate and the required intermediate certificates to SLB.

The format of the certificate chain must meet the following requirements:

- The server certificate must be put first and the content of the one or more required intermediate certificates must be put underneath without blank lines between the certificates.
- The certificate content cannot contain spaces.
- Blank lines are not allowed between the certificates. Each line must contain 64 characters. For more information, see RFC1421.
- Certificates must meet the corresponding format requirements. In most cases, the intermediate CA provides instructions about the certificate format when certificates are issued. The certificates must meet the format requirements.

The following section provides a sample certificate chain:

```
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
```

Public keys of certificates

SLB supports the following public key algorithms:

- RSA 1024
- RSA 2048
- RSA 4096
- ECDSA P-256
- ECDSA P-384
- ECDSA P-521

RSA private keys

When you upload a server certificate, you must upload the private key of the certificate.

An RSA private key must meet the following format requirements:

- The private key must start with `-----BEGIN RSA PRIVATE KEY-----` and end with -----END RSA PRIVATE KEY-----, and these parts must also be uploaded.
- Blank lines are not allowed in the content. Each line (except the last line) must contain 64 characters. The last line can contain 64 or fewer characters. For more information, see RFC1421.

You may use an encrypted private key. For example, the private key starts with `----BEGIN PRIVATE KEY-----` and ends with -----END PRIVATE KEY-----, or starts with `-----BEGIN ENCRYPTED PRIVATE KEY-----` and ends with -----END ENCRYPTED PRIVATE KEY-----. The private key may also contain `Proc-Type: 4,ENCRYPTED`. In this case, you must first run the following command to convert the private key:

```
openssl rsa -in old_server_key.pem -out new_server_key.pem
```

20.1.8.3. Upload certificates

This topic describes how to create and upload certificates to Server Load Balancer (SLB). Before you create an HTTPS listener, you must upload the required server certificate and CA certificate to SLB. You do not need to configure certificates on backend servers after you upload the certificates to SLB.

Prerequisites

- A server certificate is purchased.

- A CA certificate and a client certificate are generated.

Context

Note that you can create up to 100 certificates with each account.

Procedure

1. In the left-side navigation pane, click **Certificates**.
2. On the Certificates page, click **Create Certificate**.
3. In the **Create Certificate** panel, set the required parameters and click **Create**.

Parameter	Description
Certificate Name	Enter a name for the certificate. The name must be 1 to 80 characters in length, and can contain only letters, digits, hyphens (-), forward slashes (/), periods (.), underscores (_), and asterisks (*).
Organization	The organization to which the certificate belongs.
Resource Group	The resource set to which the certificate belongs.
Certificate Standard	Select the type of certificate. You can select **International Standard Certificate** or **National Standard Certificate**.
Public Key Certificate	The content of the server certificate. Paste the content into the editor. Click **Example** to view the valid certificate formats. For more information, see Certificate requirements.
Private Key	The private key of the server certificate. Paste the private key into the editor. Click **Example** to view the valid certificate formats. For more information, see Certificate requirements. **Notice** A private key is required only when you upload a server certificate.
Region	The region where you want to deploy the certificate.

4. Click **Create**.

20.1.8.4. Generate a CA certificate

When you configure an HTTPS listener, you can use a self-signed CA certificate. This topic describes how to generate a CA certificate and use the CA certificate to sign a client certificate.

Generate a CA certificate by using Open SSL

1. Run the following commands to create a *ca* folder in the */root* directory and then create four subfolders under the *ca* folder.

```
sudo mkdir ca
cd ca
sudo mkdir newcerts private conf server
```

   - *newcerts* is used to store the digital certificate signed by the CA certificate.
   - *private* is used to store the private Key of the CA certificate.

- *conf* is used to store the configuration files used for simplifying parameters.
- *server* is used to store the server certificate.

2. Create an *openssl.conf* file that contains the following information in the *conf* directory.

```
[ ca ]
default_ca = foo
[ foo ]
dir = /root/ca
database = /root/ca/index.txt
new_certs_dir = /root/ca/newcerts
certificate = /root/ca/private/ca.crt
serial = /root/ca/serial
private_key = /root/ca/private/ca.key
RANDFILE = /root/ca/private/.rand
default_days = 365
default_crl_days= 30
default_md = md5
unique_subject = no
policy = policy_any
[ policy_any ]
countryName = match
stateOrProvinceName = match
organizationName = match
organizationalUnitName = match
localityName = optional
commonName      = supplied
emailAddress    = optional
```

3. Run the following command to generate a private Key.

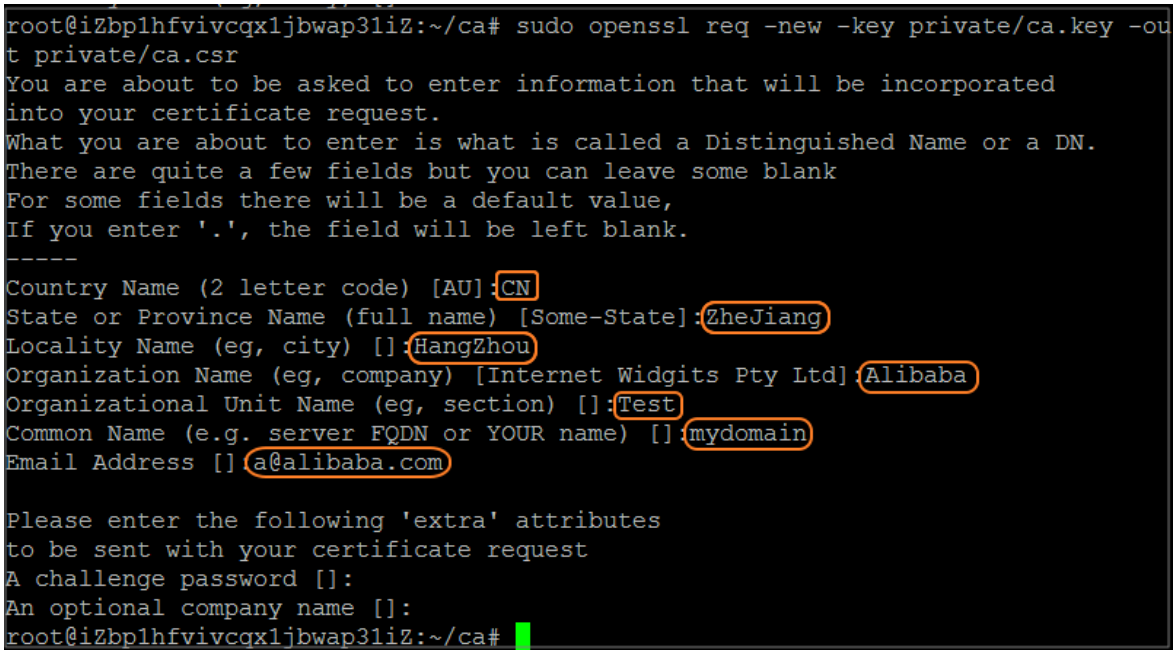
```
cd /root/ca
sudo openssl genrsa -out private/ca.key
```

The following figure is an example of the key generation.

```
root@iZbp1hfvivcqx1jbwap31iZ:~/ca/conf# cd /root/ca
root@iZbp1hfvivcqx1jbwap31iZ:~/ca# sudo openssl genrsa -out private/ca.key
Generating RSA private key, 2048 bit long modulus
.....+++
..+++
e is 65537 (0x10001)
```

4. Run the following command and input the required information according to the prompts. Press Enter to generate a *csr* file.

```
sudo openssl req -new -key private/ca.key -out private/ca.csr
```



> **Note**
>
> Common Name is the domain name of the SLB instance.

5. Run the following command to generate a *crt* file:

```
sudo openssl x509 -req -days 365 -in private/ca.csr -signkey private/ca.key -out private/ca.crt
```

6. Run the following command to set the start sequence number for the private Key, which can be any four characters.

```
sudo echo FACE > serial
```

7. Run the following command to create a CA Key library:

```
sudo touch index.txt
```

8. Run the following command to create a certificate revocation list for removing the client certificate:

```
sudo openssl ca -gencrl -out /root/ca/private/ca.crl -crldays 7 -config "/root/ca/conf/openssl.conf"
```

The output is:

```
Using configuration from /root/ca/conf/openssl.conf
```

Sign the client certificate

1. Run the following command to generate a *users* folder under the *ca* directory to store the client Key.

```
sudo mkdir users
```

2. Run the following command to create a Key for the client certificate:

```
sudo openssl genrsa -des3 -out /root/ca/users/client.key 1024
```

> **Note**
>
> Enter a pass phrase when creating the Key. It is the password to protect the private Key from unauthorized access. Enter the same password twice.

3. Run the following command to create a *csr* file for the client Key.

```
sudo openssl req -new -key /root/ca/users/client.key -out /root/ca/users/client.csr
```

Enter the pass phrase set in the previous step and other required information when prompted.

> **Note**
>
> A challenge password is the password of the client certificate. Note that it is not the password of the client Key.

4. Run the following command to sign the client Key.

```
sudo openssl ca -in /root/ca/users/client.csr -cert /root/ca/private/ca.crt -keyfile /root/ca/private/ca.key -out /root/ca/users/client.crt -config "/root/ca/conf/openssl.conf"
```

Enter *y* twice when prompted to confirm the operation.

```
root@iZbp1hfvivcqx1jbwap31iZ:~/ca# sudo openssl ca -in /root/ca/users/client.csr
 -cert /root/ca/private/ca.crt -keyfile /root/ca/private/ca.key -out /root/ca/us
ers/client.crt -config "/root/ca/conf/openssl.conf"
Using configuration from /root/ca/conf/openssl.conf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'CN'
stateOrProvinceName   :ASN.1 12:'ZheJiang'
localityName          :ASN.1 12:'HangZhou'
organizationName      :ASN.1 12:'Alibaba'
organizationalUnitName:ASN.1 12:'Test'
commonName            :ASN.1 12:'mydomain'
emailAddress          :IA5STRING:'a@alibaba.com'
Certificate is to be certified until Jun  4 15:28:55 2018 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
root@iZbp1hfvivcqx1jbwap31iZ:~/ca#
```

5. Run the following command to convert the certificate to a *PKCS12* file.

```
sudo openssl pkcs12 -export -clcerts -in /root/ca/users/client.crt -inkey /root/ca/users/client.key -out /root/ca/users/client.p12
```

Follow the prompts to enter the pass phrase of client Key. Then enter the password used for exporting the client certificate. This password is used to protect the client certificate, which is required when you install the client certificate.

6. Run the following commands to view the generated client certificate:

```
cd users
ls
```

20.1.8.5. Convert the certificate format

Server Load Balancer (SLB) supports PEM certificates only. Certificates in other formats must be converted to the PEM format before they can be uploaded to SLB. We recommend that you use Open SSL for conversion.

Convert DER to PEM

DER: This format is usually used on a Java platform. The certificate file suffix is generally *. der*, *. cer*, or *. crt*.

- Run the following command to convert the certificate format:

```
openssl x509 -inform der -in certificate.cer -out certificate.pem
```

- Run the following command to convert the private key:

```
openssl rsa -inform DER -outform PEM -in privatekey.der -out privatekey.pem
```

Convert P7B to PEM

P7B: This format is usually used in a Windows server and Tomcat.

Run the following command to convert the certificate format:

```
openssl pkcs7 -print_certs -in incertificate.p7b -out outcertificate.cer
```

Convert PFX to PEM

PFX: This format is usually used in a Windows server.

- Run the following command to extract the certificate:

```
openssl pkcs12 -in certname.pfx -nokeys -out cert.pem
```

- Run the following command to extract the private key:

```
openssl pkcs12 -in certname.pfx -nocerts -out key.pem -nodes
```

20.1.8.6. Replace a certificate

This topic describes how to replace a certificate with a new certificate. We recommend that you replace certificates before they expire to avoid impacts on your service.

Procedure

1. Create and upload a new certificate.

   For more information, see Overview.
2. Configure the certificate for the target HTTPS listener.

   For more information, see Add an HTTPS listener.
3. On the **Certificates** page, find the certificate to be replaced and click **Delete** in the Actions column.
4. In the dialog box that appears, click **OK**.

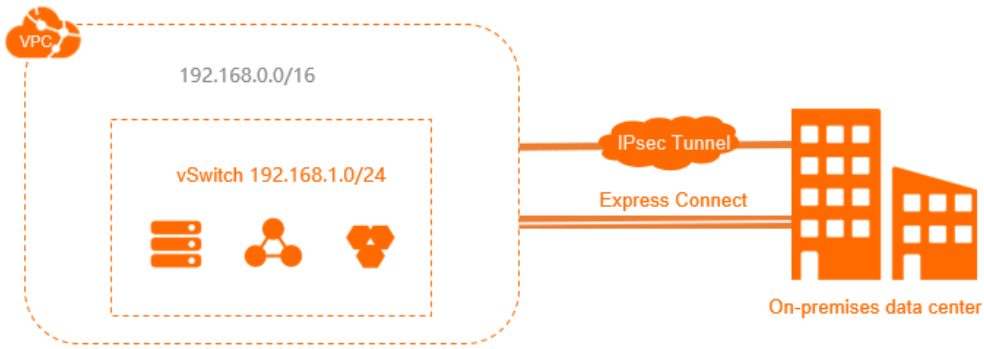
21.Virtual Private Cloud (VPC)

21.1. User Guide

21.1.1. What is a VPC?

A virtual private cloud (VPC) is a private network dedicated for your use. You have full control over your VPC. For example, you can specify the CIDR block and configure route tables and gateways. In a VPC, you can deploy Apsara Stack resources, such as Elastic Compute Service (ECS) instances, ApsaraDB RDS instances, and Server Load Balancer (SLB) instances.

Furthermore, you can connect your VPC to other VPCs or on-premises networks to create a custom network environment. This way, you can migrate applications to the cloud and extend data centers.



Components

Each VPC consists of one vRouter, at least one private CIDR block, and at least one vSwitch.

- Private CIDR blocks

  When you create a VPC and a vSwitch, you must specify the private IP address range for the VPC in CIDR notation.

  You can use the standard private CIDR blocks listed in the following table and their subsets as CIDR blocks for your VPCs. For more information, see the Plan and design a VPC topic in *User Guide*.

CIDR blocks	Number of available private IP addresses (system reserved ones excluded)
192.168.0.0/16	65,532
172.16.0.0/12	1,048,572
10.0.0.0/8	16,777,212

- vRouters

  A vRouter is the hub of a VPC and serves as a gateway between the VPC and other networks. After a VPC is created, a vRouter is automatically created for the VPC. Each vRouter is associated with a route table.

  For more information, see the Route table overview topic in *User Guide*.

- vSwitches

A vSwitch is a basic network component that connects different cloud resources in a VPC. After you create a VPC, you can create a vSwitch to divide your VPC into multiple subnets. vSwitches deployed in a VPC can communicate with each other over the private network. You can deploy your applications in vSwitches that belong to different zones to improve service availability.

For more information, see the Create a vSwitch topic in *User Guide*.

21.1.2. Log on to the VPC console

This topic describes how to log on to the Virtual Private Cloud (VPC) console of Apsara Uni-manager by using the Google Chrome browser.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Login**.
4. In the top navigation bar, choose **Products > Networking > Virtual Private Cloud**.

21.1.3. Quick start

21.1.3.1. Plan and design a VPC

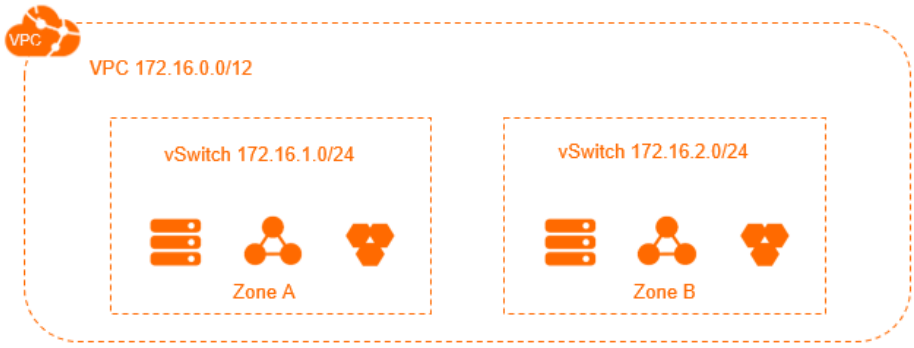
Before you create virtual private clouds (VPCs) and VSwitches, you need to plan the quantity and Classless Inter-domain Routing (CIDR) blocks of VPCs and VSwitches.

- How many VPCs are required?
- How many VSwitches are required?
- How do I specify CIDR blocks?
- How do I specify CIDR blocks if I want to connect a VPC to other VPCs or on-premises data centers?

How many VPCs are required?

- One VPC

  We recommend that you create one VPC if you do not need to deploy systems in multiple regions or separate VPCs.

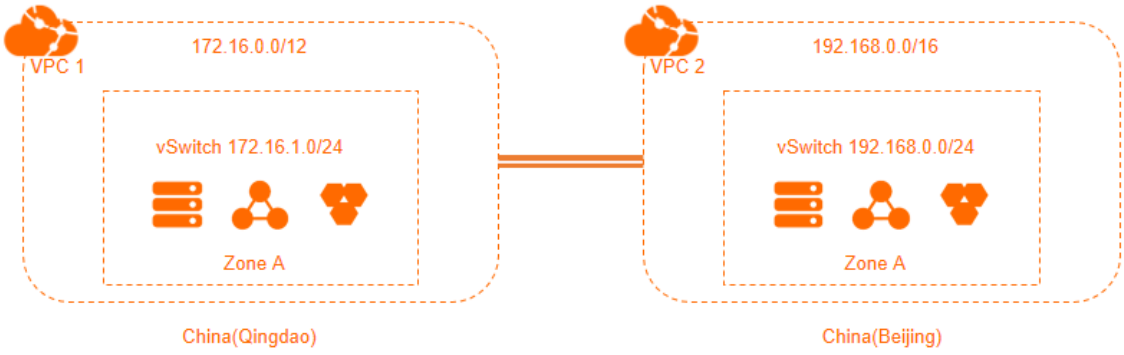


- Multiple VPCs

  We recommend that you create multiple VPCs if you need to:

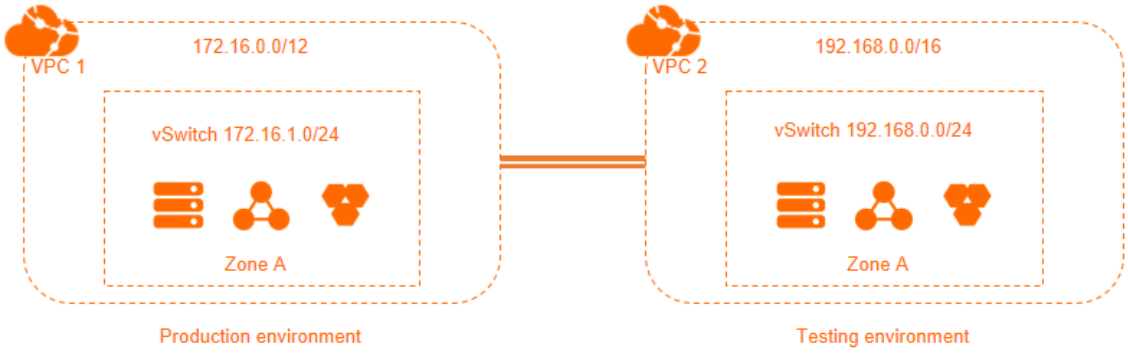
  - Deploy application systems across regions.

    A VPC cannot be deployed across regions. If you want to deploy your application systems in different regions, you must create multiple VPCs. You can use Express Connect and VPN Gateway to connect VPCs.



  - Separate IT systems

    To separate IT systems, you must create multiple VPCs. The following figure shows an example of isolating a production environment from a test environment by deploying them in separate VPCs.



How many VSwitches are required?

We recommend that you create at least two VSwitches for each VPC and deploy these VSwitches in different zones to achieve zone-disaster recovery.

After you deploy your applications in different zones within a region, you must measure the network latency between these applications. This is because the cross-zone network latency may be higher than expected due to complex data processing or cross-zone calls. An ideal approach is to optimize and adjust your systems to strike a balance between availability and latency.

In addition, the sizes and designs of your IT systems must also be taken into consideration when you create VSwitches. If you allow traffic from the Internet to be routed to and from the frontend systems, you can deploy the front-end systems in different VSwitches and the backend systems in other VSwitches to create a robust disaster recovery strategy.

How do I specify CIDR blocks?

When you create VPCs and VSwitches, you must specify their private IP address ranges in the form of CIDR blocks.

- VPC CIDR blocks

  You can use 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, or their subsets as the CIDR blocks of your VPCs. To specify CIDR blocks for VPCs, follow these rules:

  - If you have only one VPC and this VPC does not need to communicate with any on-premises data center, you can use one of the preceding CIDR blocks or one of their subsets as the CIDR block of the VPC.
  - If you have multiple VPCs, or you need to build a hybrid cloud to integrate VPCs and on-premises data centers, we recommend that you use the subsets of the preceding CIDR blocks for your VPCs. In this case, the mask cannot be longer than 16 bits.

- VSwitch CIDR blocks

  The CIDR block of a VSwitch must be a subset of the CIDR block of the VPC this VSwitch resides in. For example, if the CIDR block of a VPC is 192.168.0.0/16, the CIDR block of a VSwitch in the VPC must be a segment from 192.168.0.0/17 to 192.168.0.0/29.

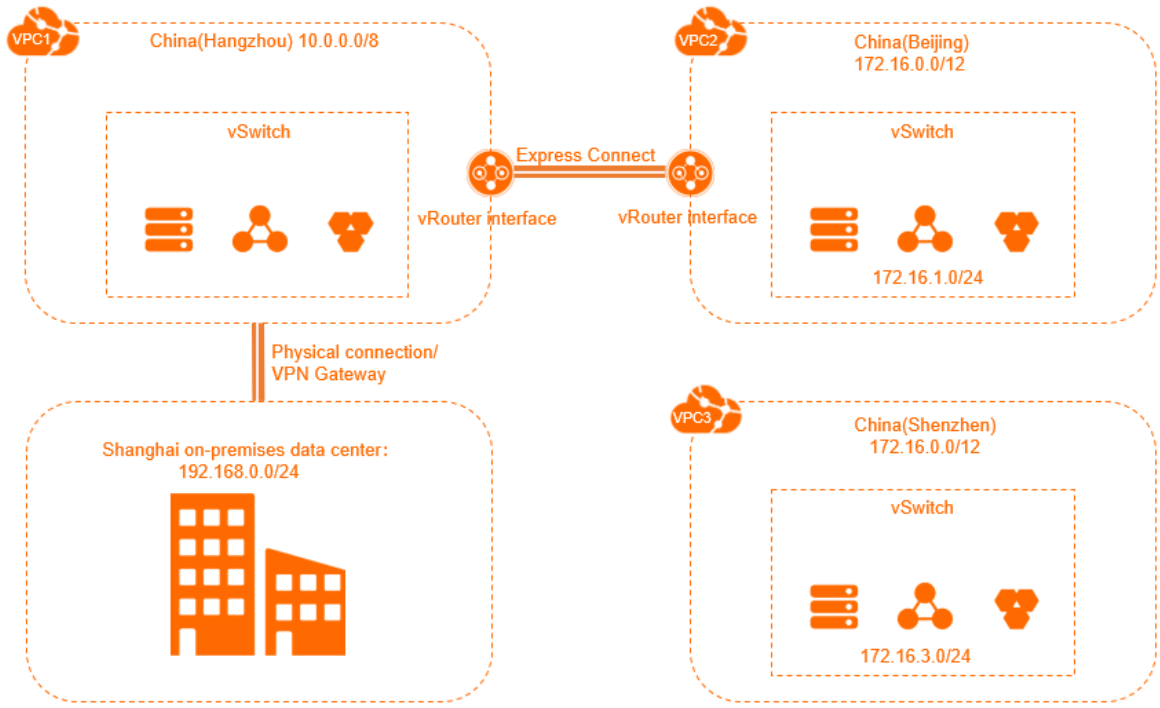
  To specify CIDR blocks for VSwitches, follow these rules:

  - The CIDR block size for a VSwitch is between a 16-bit mask and a 29-bit mask. It means that 8 to 65,536 IP addresses can be provided. This range is set because a 16-bit host address space provides addressing for 65,534 ECS instances, which can meet your needs in most cases, while a mask smaller than 29 bits can only allow very few usable host addresses.
  - The first and the last three IP addresses in each VSwitch CIDR block are reserved by the system. For example, if the CIDR block of a VSwitch is 192.168.1.0/24, the IP addresses 192.168.1.0, 192.168.1.253, 192.168.1.254, and 192.168.1.255 are reserved.
  - You must check the number of ECS instances in the VSwitch before you specify the CIDR block of a VSwitch.

How do I specify CIDR blocks if I want to connect a VPC to other VPCs or on-premises data centers?

Before you connect your VPC to another VPC or an on-premises data center, you must make sure that the CIDR block of your VPC does not conflict with that of the peer network.

For example, assume you have three VPCs: VPC1 in China (Hangzhou), VPC2 in China (Beijing), and VPC3 in China (Shenzhen), as shown in the following figure. Express Connect circuit is used for VPC1 and VPC2 to communicate with each other. VPC3 does not communicate with other VPCs, but may need to communicate with VPC2 in the future. Additionally, you have an on-premises data center in Shanghai, and you need to connect it to VPC1 by using an Express Connect circuit.



In this example, the CIDR block of VPC2 is different from the CIDR block of VPC1, but is the same with the CIDR block of VPC3. However, considering that VPC2 and VPC3 may need to communicate with each other later in the private network, the VSwitches in these VPCs are assigned with different CIDR blocks. This example demonstrates that VPCs communicating with each other can have identical CIDR blocks, but their VSwitches must have different CIDR blocks.

When you specify CIDR blocks for multiple VPCs that need to communicate with each other, follow these rules:

- The preferred practice is to specify different CIDR blocks for different VPCs. You can use the subsets of the standard CIDR blocks to increase the number of available CIDR blocks.
- If you cannot assign different CIDR blocks for VPCs, try to specify different CIDR blocks for the VSwitches in these VPCs.
- If you cannot assign different CIDR blocks for all VSwitches in these VPCs, make sure that different CIDR blocks are configured for the VSwitches communicating with each other.

21.1.3.2. Create an IPv4 VPC

This topic describes how to create a virtual private cloud (VPC) with an IPv4 CIDR block and create an Elastic Compute Service (ECS) instance in the VPC.

Prerequisites

To deploy cloud resources in a VPC, you must have network subnetting prepared first. For more information, see Plan networks.

Step 1: Create a VPC

Perform the following steps to create a VPC:

1. Log on to the VPC console.
2. On the **VPCs** page, click **Create VPC**.
3. On the **Create VPC** page, set the following parameters and click **Submit**.

Parameter	Description
Organization	Select the organization to which the VPC belongs.
Resource Set	Select the resource set to which the VPC belongs.
Region	Select the region where you want to deploy the VPC.
Sharing Scope	Select the participants who can use the VPC to create resources. ○ **Current Resource Set**: Only the administrator of the current resource set can use the VPC to create resources. ○ **Current Organization and Subordinate Organization**: Only the administrators of the current organization and its subordinate organization can use the VPC to create resources. ○ **Current Organization**: Only the administrator of the current organization can use the VPC to create resources. **Current Resource Set** is selected in this example.
VPC Name	Enter a name for the VPC. The name must be 2 to 128 characters in length, and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`. **VPCtest** is used in this example.
IPv4 CIDR Block	Select an IPv4 CIDR block for the VPC. The following setting methods are supported: ○ **Recommended CIDR Block**: You can use one of the following standard IPv4 CIDR blocks: 192.168.0.0/16, 172.16.0.0/12, and 10.0.0.0/8. ○ **Custom CIDR Block**: You can use 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, or a subset of these CIDR blocks. The subnet mask must be 8 to 28 bits in length. For example, enter 192.168.0.0/16. In this example, 192.168.0.0/16 is used as the IPv4 CIDR block of the VPC. **Note** After you create a VPC, you cannot change its IPv4 CIDR block.
IPv6 CIDR Block	Specify whether to assign an IPv6 CIDR block. ○ **Do Not Assign**: No IPv6 CIDR block will be assigned to the VPC. ○ **Assign**: An IPv6 CIDR block will be automatically assigned to the VPC. **Do Not Assign** is selected in this example.
Description	Enter a description for the VPC. The description must be 2 to 256 characters in length, and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.

4. Click **Back to Console**. On the **VPCs** page, you can view the VPCs that are created.

Step 2: Create a vSwitch

Perform the following steps to create a vSwitch in a VPC:

1. In the left-side navigation pane, click **vSwitches**.
2. On the **vSwitches** page, click **Create vSwitch**.
3. On the **Create vSwitch** page, set the following parameters and click **Submit**.

Parameter	Description
Organization	Select the organization to which the vSwitch belongs.
Resource Set	Select the resource set to which the vSwitch belongs.
Region	Select the region where the vSwitch is deployed.
Zone	Select the zone to which the vSwitch belongs. In a VPC, a vSwitch can belong to only one zone. However, you can deploy cloud resources in vSwitches that reside in different zones to achieve cross-zone disaster recovery. **Note** A cloud instance can be deployed in only one vSwitch.
Sharing Scope	Select the participants that can use the vSwitch to create resources. ○ **Current Resource Set**: Only the administrator of the current resource set can use the vSwitch to create resources. ○ **Current Organization and Subordinate Organization**: Only the administrators of the current organization and its subordinate organization can use the vSwitch to create resources. ○ **Current Organization**: Only the administrator of the current organization can use the vSwitch to create resources. **Current Resource Set** is selected in this example.
VPC	The VPC in which you want to create the vSwitch. VPCtest is selected in this example.
Dedicated for Out-of-cloud Physical Machines	Specify whether the vSwitch to be created is dedicated for bare-metal servers. For more information about bare-metal servers, see the **VPC bare-metal server features** topic in *BMS user guide*. **No** is selected in this example.
vSwitch Name	Enter a name for the vSwitch. The name must be 2 to 128 characters in length, and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.
IPv4 CIDR Block	Specify an IPv4 CIDR block for the vSwitch. The default IPv4 CIDR block is used in this example.

Parameter	Description
IPv6 CIDR Block	Specify an IPv6 CIDR block for the vSwitch. **Do Not Assign** is selected in this example.
Description	Enter a description for the vSwitch. The name must be 2 to 256 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.

Step 3: Create a security group

Perform the following steps to create a security group:

1. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.
2. Choose **Networks and Security > Security Groups**.
3. On the **Security Groups** page, click **Create Security Group**.
4. On the **Create Security Group** page, set the following parameters and click **Submit**.

Parameter	Description
Organization	Select the organization to which the security group belongs.
Resource Set	Select the resource set to which the security group belongs.
Region	Select the region to which the security group belongs. Make sure that the security group and the VPC belong to the same region.
Zone	Select the zone to which the security group belongs.
VPC	The VPC to which the security group belongs.
Security Group Name	Enter a name for the security group. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.
Description	Enter a description for the security group. The description must be 2 to 256 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.

Step 4: Create an ECS instance

Perform the following steps to create an ECS instance in the VPC:

1. In the top navigation bar, choose **Products > Networking > Virtual Private Cloud**.
2. In the left-side navigation pane, click **vSwitches**.
3. In the top navigation bar, select the region where the vSwitch is deployed.
4. On the **vSwitches** page, find the vSwitch and choose **Create > ECS Instance** in the **Actions** column.
5. On the **Create ECS Instance** page, set the parameters and click **Submit**.

   For more information about how to configure ECS instances, see **Create an instance** in **Quick start** of *ECS u*

ser guide.

21.1.3.3. Create an IPv6 VPC

This topic describes how to create a virtual private cloud (VPC) that supports IPv6 CIDR blocks and then create an Elastic Compute Service (ECS) instance that is assigned an IPv6 address in the VPC to access IPv6 services.

Step 1: Create a VPC and a vSwitch

Before you deploy cloud resources in a VPC, you must create a VPC and a vSwitch.

Perform the following steps to create a VPC and a vSwitch:

1. Log on to the VPC console.
2. On the **VPCs** page, click **Create VPC**.
3. On the **Create VPC** page, set the following parameters to configure the VPC and click **OK**.

Parameter	Description
Organization	Select the organization to which the VPC belongs.
Resource Set	Select the resource set to which the VPC belongs.
Region	Select the region where you want to deploy the VPC.
Sharing Scope	Specify the scope of entities that are allowed to use the VPC. ○ **Current Resource Set**: If you select this option, the administrator of the current resource set can create resources in the VPC. ○ **Current Organization and Subordinate Organizations**: If you select this option, administrators that belong to the current organization and subordinate organizations can create resources in the VPC. ○ **Current Organization**: If you select this option, administrators that belong to the current organization can create resources in the VPC. In this example, **Current Resource Set** is selected.
VPC Name	Enter a name for the VPC. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character and cannot start with `http://` or `https://`. In this example, **VPCtest** is entered.
IPv4 CIDR Block	Specify the IPv4 CIDR block of the VPC. You can specify an IPv4 CIDR block in one of the following ways: ○ **Recommended CIDR Block**: You can use one of the following standard IPv4 CIDR blocks: 192.168.0.0/16, 172.16.0.0/12, and 10.0.0.0/8. ○ **Custom CIDR Block**: Enter 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, or a subset of these CIDR blocks. The subnet mask must be 8 to 28 bits in length. For example, you can enter 192.168.0.0/16. In this example, Recommended CIDR Block is selected and 192.168.0.0/16 is selected as the IPv4 CIDR block of the VPC. **Note** After you create a VPC, you cannot change its IPv4 CIDR block.

Parameter	Description
IPv6 CIDR Block	Specify whether to assign an IPv6 CIDR block. ○ **Do Not Assign**: If you select this option, the system does not assign an IPv6 CIDR block to the VPC. ○ **Assign**: If you select this option, the system automatically assigns an IPv6 CIDR block to the VPC. In this example, **Assign** is selected.
Description	Enter a description for the VPC. The description must be 2 to 256 characters in length and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character. It cannot start with `http://` or `https://`.

4. Click **Back to Console**. In the left-side navigation pane, click **VSwitches**.
5. On the **VSwitches** page, click **Create VSwitch**.
6. On the **vSwitch** page, set the following parameter to configure the vSwitch and click **Submit**.

Parameter	Description
Organization	Select the organization to which the vSwitch belongs.
Resource Set	Select the resource set to which the vSwitch belongs.
Region	Select the region where you want to deploy the vSwitch.
Zone	Select the zone where you want to deploy the vSwitch. In a VPC, each vSwitch can be deployed in only one zone. You cannot deploy a vSwitch across zones. However, you can deploy cloud resources in vSwitches that belong to different zones to achieve zone-disaster recovery. **Note** Each cloud resource can be added to only one vSwitch.

Parameter	Description
Sharing Scope	Specify the scope of entities that are allowed to use the vSwitch. ○ **Current Resource Set**: If you select this option, the administrator of the current resource set can create resources in the vSwitch. ○ **Current Organization and Subordinate Organizations**: If you select this option, administrators that belong to the current organization and subordinate organizations can create resources in the vSwitch. ○ **Current Organization**: If you select this option, administrators that belong to the current organization can can create resources in the vSwitch. In this example, **Current Resource Set** is selected.
VPC	Select the VPC where you want to deploy the vSwitch. In this example, VPCtest is selected.
Dedicated for Out-of-cloud Physical Machines	Specify whether the vSwitch to be created is dedicated to bare-metal servers. For more information about bare-metal servers, see the **Bare-metal servers in VPCs** topic in *Bare-metal Server Management Service User Guide*. In this example, **No** is selected.
vSwitch Name	Enter a name for the vSwitch. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character. It cannot start with `http://` or `https://`.
IPv4 CIDR Block	Enter an IPv4 CIDR block for the vSwitch. In this example, the default IPv4 CIDR block is used.
IPv6 CIDR Block	Enter an IPv6 CIDR block for the vSwitch. In this example, the default IPv6 CIDR block is used.
Description	Enter a description for the vSwitch. The description must be 2 to 256 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character. It cannot start with `http://` or `https://`.

Step 2: Create a security group

Perform the following steps to create a security group:

1. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.
2. Choose **Networks and Security > Security Groups**.
3. On the **Security Groups** page, click **Create Security Group**.
4. On the **Create Security Group** page, set the following parameters to configure the security group and click **Submit**.

Parameter	Description
Organization	Select the organization to which the security group belongs.

Parameter	Description
Resource Set	Select the resource set to which the security group belongs.
Region	Select the region to which the security group belongs. Make sure that the security group and the VPC belong to the same region.
Zone	Select the zone to which the security group belongs.
VPC	Select the VPC to which the security group belongs.
Security Group Name	Enter a name for the security group. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character. It cannot start with `http://` or `https://`.
Description	Enter a description for the security group. The description must be 2 to 256 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character. It cannot start with `http://` or `https://`.

Step 3: Create and configure an ECS instance

After you create a VPC and a vSwitch, you must create an ECS instance and assign an IPv6 address to the ECS instance. You must associate this IPv6 address with the network interface controller (NIC) of the ECS instance.

Perform the following steps to create and configure an ECS instance:

1. In the top navigation bar, choose **Products > Networking > Virtual Private Cloud**.
2. In the left-side navigation pane, click **VSwitches**.
3. Select the region where the vSwitch is created.
4. On the **VSwitches** page, find the vSwitch that you want to manage and choose **Create > ECS Instance** in the **Actions** column.
5. On the **Create ECS Instance** page, configure the ECS instance and click **Submit**.

   In this example, **Assign** is selected. Therefore, an IPv6 IP address is assigned to the ECS instance. For more information about other parameters that you are required to specify when you create an ECS instance, see **Create an ECS instance** in **Quick Start** of *Elastic Compute Service User Guide*.
6. Return to the **Instances** page and click the instance ID to view the IPv6 address that is assigned to the ECS instance.
7. Configure a static IPv6 address.
   - If the image of your ECS instance supports DHCPv6, you do not need to manually configure a static IPv6 address. DHCPv6 enables automatic configuration of IPv6 addresses. Therefore, if your ECS instance image supports DHCPv6, the ECS instance can use the assigned IPv6 address to communicate within the private network.

     The following images support DHCPv6:

     - Linux images:
       - CentOS 7.6 IPV6 64Bit
       - CentOS 6.10 64Bit
       - SUSE Linux Enterprise Server 12 SP4 64Bit

- Windows Server images

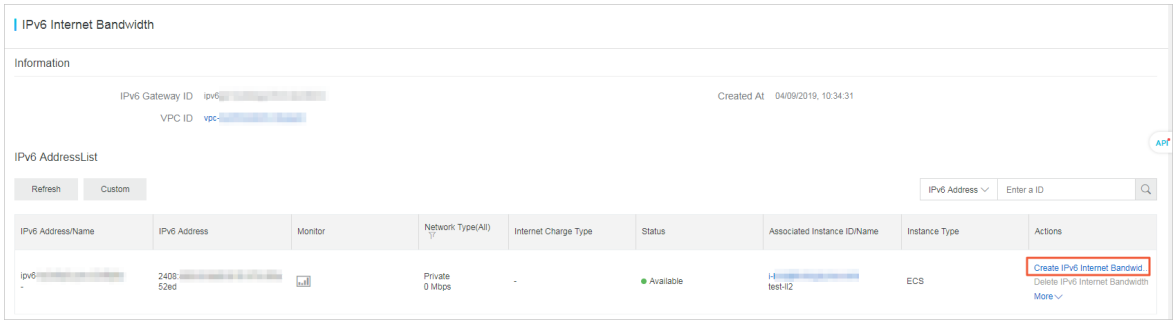
- If the image of your ECS instance does not support DHCPv6, you must manually configure an IPv6 address for the ECS instance. We recommend that you refer to the related documentation for each image for configuration guidance.

Step 4: Purchase an IPv6 Internet bandwidth plan

By default, IPv6 addresses are only used for communication within private networks. If you want to allow an instance that is assigned an IPv6 address to access the Internet or receive requests from IPv6 clients over the Internet, you must purchase an Internet bandwidth plan for the IPv6 address.

Perform the following steps to purchase an Internet bandwidth plan for the IPv6 address:

1. In the top navigation bar, choose **Products > Networking > IPv6 Gateway**.
2. Select the region where the IPv6 gateway is created.
3. On the **IPv6 Gateway** page, find the IPv6 gateway that you want to manage and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **IPv6 Internet Bandwidth**.
5. On the **IPv6 Internet Bandwidth** page, find the IPv6 address that you want to manage and click **Enable IPv6 Internet Bandwidth** in the **Actions** column.



6. Select a bandwidth plan and click **OK**.

   The maximum IPv6 Internet bandwidth for an IPv6 gateway of the Free, Enterprise, or Enhanced Edition is 2 Gbit/s.

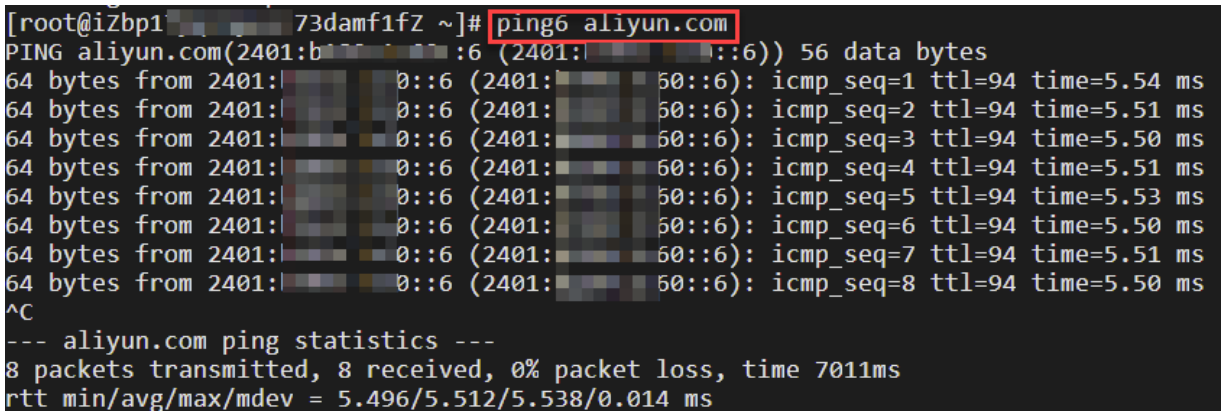
Step 5: Configure security group rules

IPv4 and IPv6 addresses are independent of each other. If the current security group rules do not apply to your IPv6 services, you must configure security group rules for the ECS instances to regulate communication with IPv6 addresses.

For more information about how to configure security rules, see the **Add security group rules** chapter in **Security Groups** of *Elastic Compute Service User Guide*.

Step 6: Test the network connectivity

Log on to an ECS instance and **ping** an IPv6 service to test the network connectivity.



21.1.4. VPCs and VSwitches

21.1.4.1. Overview

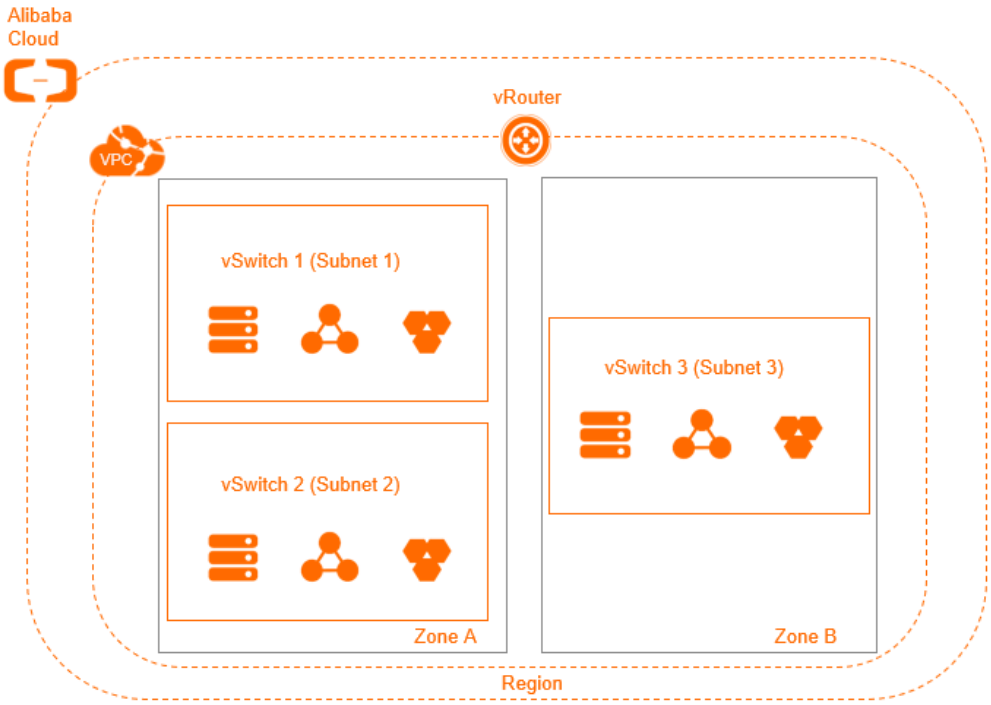
To get started with Virtual Private Cloud, you need to create at least one virtual private cloud (VPC) and one VSwitch. You can create VSwitches in a VPC to partition a VPC into multiple subnets. By default, subnets (VSwitches) within a VPC can communicate with each other over the private network.

VPCs and VSwitches

A VPC is a virtual network dedicated for your use. You can deploy cloud resources in VPCs that you define.

> **Note** A cloud resource cannot be directly deployed in a VPC, but can be deployed in a VSwitch of the VPC.

A VSwitch is a basic network component in a VPC and is used to connect cloud resources. Each VPC must reside entirely within one region and cannot span multiple regions. However, a VPC spans all of the zones in a region, which means you can create one or more VSwitches in each zone to partition a VPC into subnets.



CIDR blocks and IP addresses

VPCs support both IPv4 and IPv6 addressing protocols. By default, VPCs use the IPv4 addressing protocol. You can enable the IPv6 addressing protocol as needed.

VPCs can operate in a dual-stack mode, which allows your resources to communicate over IPv4, or IPv6, or both. IPv4 and IPv6 addresses are independent of each other. Therefore, you must configure routing and security groups in your VPC separately for IPv4 and IPv6.

The following table summarizes the differences between IPv4 and IPv6 in Apsara Stack VPC.

IPv4 VPC	IPv6 VPC
The format is 32-bit, 4 groups of up to 3 decimal digits.	The format is 128-bit, 8 groups of 4 hexadecimal digits.
The IPv4 addressing protocol is enabled for all VPCs by default.	The IPv6 addressing protocol is optional for a VPC.
The VPC Classless Inter-domain Routing (CIDR) block size can be from /8 to /24.	The VPC CIDR block size is fixed at /61.
The VSwitch CIDR block size can be from /16 to /29.	The VSwitch CIDR block size is fixed at /64.
You can choose the private IPv4 CIDR block for your VPC.	Apsara Stack automatically assigns an IPv6 CIDR block for your VPC from its IPv6 address pool.You cannot select your own range.

IPv4 VPC	IPv6 VPC
Supported on all instance types.	Not supported on certain instance types. For more information, see **Instance types** under **What is ECS** in the *Apsara Stack Elastic Compute Service User Guide*.
Elastic IPv4 addresses are supported.	Elastic IPv6 addresses are not supported.
VPN gateways and NAT gateways are supported.	VPN gateways and NAT gateways are supported.

By default, the IPv4 and IPv6 addresses provided for VPCs can only be used for communication within the private network. Resources in different VSwitches within a VPC communicate with one another over private network connections. To connect a VPC to another VPC or an on-premises data center, you need to configure Express Connect or VPN Gateway. .

To enable cloud resources in a VPC to access the Internet, set the following configurations:

- IPv4 communication

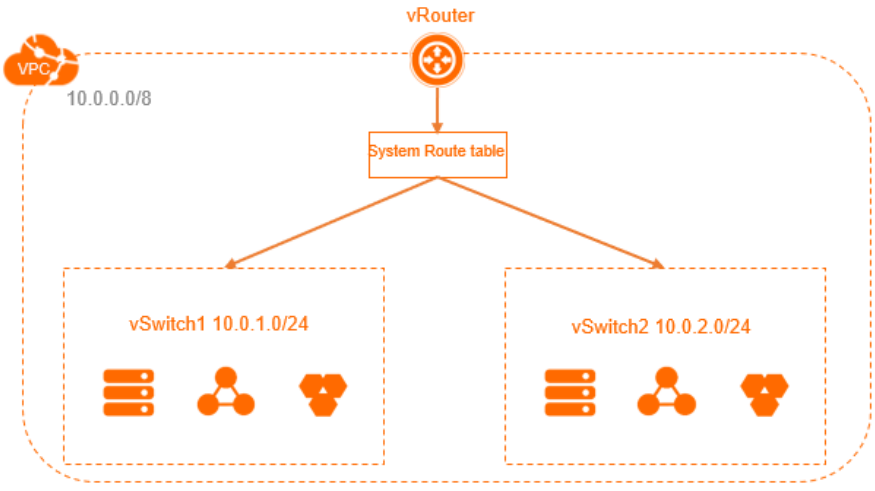
  You can configure a NAT gateway or associate elastic IP addresses (EIPs) to the Elastic Compute Service (ECS) instances in a VPC to allow these ECS instances to access the Internet by using IPv4 addresses.

- IPv6 communication

  To enable cloud resources in a VPC to access the Internet by using an IPv6 address, you must purchase public bandwidth for the IPv6 address. You can also configure an egress-only rule for an IPv6 address to allow outbound communication over IPv6 from instances in your VPC to the Internet, and prevent clients on the Internet from initiating IPv6 connections with your instances.

Routing

After a VPC is created, the system automatically creates a system route table and adds system routes to the route table for traffic management. Each VPC has only one system route table, which is generated automatically upon the creation of the VPC. You cannot create or delete system route tables.



If one destination address matches more than one route entry in a route table, the system selects an entry by implementing the longest prefix match algorithm, whereby the most specific of the matching entries, the one with the longest subnet mask, is used to route traffic. You can add a custom route entry to route traffic destined for a specific destination. For more information, see Add a custom route entry.

21.1.4.2. VPC management

21.1.4.2.1. Create a VPC

A virtual private cloud (VPC) is a private network dedicated for your use. You have full control over your VPC. For example, you can specify CIDR blocks, configure route tables and gateways for your VPC. You can deploy cloud resources in your own VPC, such as Elastic Compute Service (ECS) instances, ApsaraDB RDS instances, and Server Load Balancer (SLB) instances. This topic describes how to create a VPC.

Prerequisites

Before you create a VPC, you must plan your networks. For more information, see Plan networks.

Procedure

1. Log on to the VPC console.
2. In the top navigation bar, select the region where you want to deploy the VPC.

   > **Note** The VPC must be deployed in the same region as that of the cloud resources that you want to deploy in this VPC.

3. On the **VPC** page, click **Create VPC**.
4. On the **Create VPC** page, configure the VPC and click **OK**. The following table describes the parameters.

Parameter	Description
Organization	Select the organization to which the VPC belongs.
Resource Set	Select the resource set to which the VPC belongs.
Region	Select the region where your VPC is deployed.
Sharing Scope	Select the participants who can use the VPC to create resources. ○ **Current Resource Set**: Only the administrator of the current resource set can use the VPC to create resources. ○ **Current Organization and Subordinate Organization**: Only the administrators of the current organization and its subordinate organization can use the VPC to create resources. ○ **Current Organization**: Only the administrator of the current organization can use the VPC to create resources.
VPC Name	Enter a name for the VPC. The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.

Parameter	Description
IPv4 CIDR Block	Select an IPv4 CIDR block for the VPC. The following setting are supported: ○ **Recommended CIDR Block**: You can use one of the following standard IPv4 CIDR blocks: 192.168.0.0/16, 172.16.0.0/12, and 10.0.0.0/8. ○ **Custom CIDR Block**: You can use 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, or a subset of these CIDR blocks. The subnet mask must be 8 to 28 bits in length. For example, enter 192.168.0.0/24. **Note** After you create a VPC, you cannot change its IPv4 CIDR block.
IPv6 CIDR Block	Specify whether to assign an IPv6 CIDR block. ○ **Do Not Assign**: No IPv6 CIDR block will be assigned to the VPC. ○ **Assign**: An IPv6 CIDR block will be automatically assigned to the VPC. If you set this parameter to Assign, the system automatically creates a free IPv6 gateway for this VPC, and assigns an IPv6 CIDR block with the subnet mask /56, such as 2xx1: db8::/56. By default, IPv6 addresses can only be used to communicate within private networks. If you want to allow an instance assigned with an IPv6 address to access the Internet or be accessed by IPv6 clients over the Internet, you must purchase an Internet bandwidth plan for the IPv6 address. For more information, see the **Activate IPv6 Internet bandwidth** section of the **Manage IPv6 Internet bandwidth** topic of the *IPv6 gateway user guide*.
Description	Enter a description for the VPC. The description must be 2 to 256 characters in length and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.

21.1.4.2.2. Add a secondary IPv4 CIDR block

This topic describes how to expand a virtual private cloud (VPC) by adding a secondary IPv4 CIDR block to the VPC.

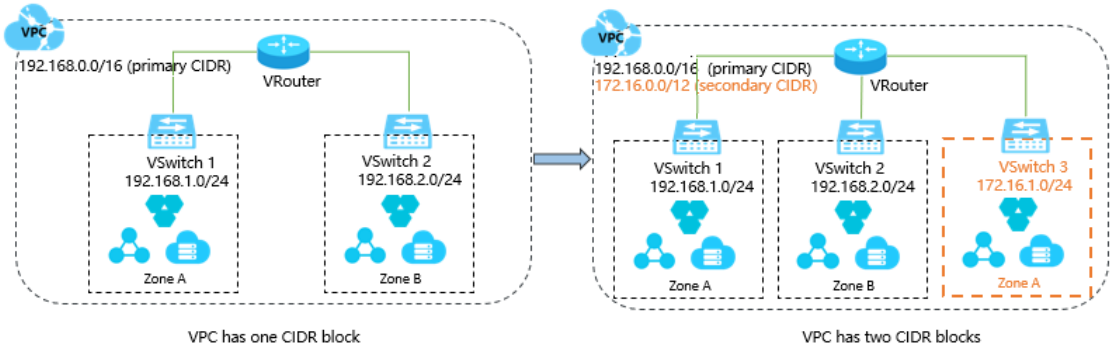
Prerequisites

A VPC is created. For more information, see Create a VPC.

Context

When you create a VPC, the IPv4 CIDR block you specified is the primary CIDR block. After the VPC is created, the primary IPv4 CIDR block of the VPC cannot be modified. However, you can add a secondary IPv4 CIDR block to expand the VPC. After you add the secondary IPv4 CIDR block, both the primary and secondary IPv4 CIDR blocks are in effect. You can create a vSwitch with the primary or a secondary IPv4 CIDR block. However, each vSwitch belongs to only one VPC CIDR block.

The system automatically adds a vSwitch route to the VPC route table when you create a vSwitch with the primary or a secondary IPv4 CIDR block. The destination CIDR block of a vSwitch route is the CIDR block with which the vSwitch is created. The CIDR block range cannot be the same as or larger than those of other routes in the route table of the VPC.



Note You can add only one secondary IPv4 CIDR block to a VPC and you cannot increase the quota.

Procedure

1. Log on to the VPC console.
2. In the top navigation bar, select the region where the VPC is deployed.
3. On the **VPC** page, find the VPC and click **Manage** in the **Actions** column.
4. On the **CIDRs** tab, click **Add IPv4 CIDR**.
5. In the **Add Secondary CIDR** panel, set the following parameters and click **OK**.

Parameter	Description
VPC	The VPC to which you want to add the secondary IPv4 CIDR block.
Secondary CIDR	Select a method to configure the secondary IPv4 CIDR block: ○ **Default CIDR Block**: You can specify one of the following standard IPv4 CIDR blocks as the secondary IPv4 CIDR block: 192.168.0.0/16, 172.16.0.0/12, and 10.0.0.0/8. ○ **Custom CIDR Block**: You can specify one of the following standard IPv4 CIDR blocks and their subnets as the secondary IPv4 CIDR block: 192.168.0.0/16, 172.16.0.0/12, and 10.0.0.0/8. When you add a secondary IPv4 CIDR block, take note of the following items: ○ The CIDR block cannot start with 0. The subnet mask must be 8 to 24 bits in length. ○ The secondary IPv4 CIDR block cannot overlap with the primary IPv4 CIDR block or or an existing secondary IPv4 CIDR block? For example, if the primary IPv4 CIDR block of a VPC is 192.168.0.0/16, you cannot specify one of the following CIDR blocks as a secondary IPv4 CIDR block: ■ A larger CIDR block that overlaps with 192.168.0.0/16, such as 192.168.0.0/8. ■ 192.168.0.0/16. ■ A smaller CIDR block that overlaps with 192.168.0.0/16, such as 192.168.0.0/24.

What's next

Work with vSwitches

21.1.4.2.3. Delete a secondary IPv4 CIDR block

This topic describes how to delete a secondary IPv4 CIDR block of a virtual private cloud (VPC). You cannot delete the primary IPv4 CIDR block of a VPC.

Prerequisites

Make sure that you have deleted the vSwitch that is created with the secondary IPv4 CIDR block. For more information, see Delete a vSwitch.

Procedure

1. Log on to the VPC console.
2. In the top navigation bar, select the region where the VPC is deployed.
3. On the **VPCs** page, find the VPC and click **Manage** in the **Actions** column.
4. On the **CIDRs** tab, find the secondary IPv4 CIDR block and click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

21.1.4.2.4. Modify the name and description of a VPC

This topic describes how to modify the name and description of a virtual private cloud (VPC).

Procedure

1. Log on to the VPC console.
2. In the top navigation bar, select the region where your VPC is deployed.
3. On the **VPCs** page, find the target VPC network and click **Manage** in the **Actions** column.
4. In the **VPC Details** section, click **Edit** next to **Name**. In the dialog box that appears, enter a new name for the VPC and click **OK**.

   The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), and hyphens (-). It must start with a letter or a Chinese character.
5. Click **Edit** next to **Description**. In the dialog box that appears, enter a new description, and click **OK**.

   The description must be 2 to 256 characters in length and cannot start with `http://` or `https://`.

21.1.4.2.5. Delete a VPC

This topic describes how to delete a virtual private cloud (VPC). After you delete a VPC, the VRouter and route tables associated with this VPC are also deleted.

Prerequisites

Before you delete a VPC, make sure that the following requirements are met:

- No VSwitch exists in the VPC. If the VPC has one or more VSwitches, we recommend that you delete the VSwitches first before deleting the VPC. For more information, see Delete a vSwitch.
- No IPv6 gateway is associated with the VPC. If the VPC is associated with an IPv6 gateway, we recommend that you delete the IPv6 gateway first before deleting the VPC.

Procedure

1. Log on to the VPC console.
2. In the top navigation bar, select the region where your VPC is deployed.
3. On the **VPCs** page, find the target VPC and click **Delete** in the **Actions** column.
4. In the **Delete VPC** dialog box, click **OK**.

21.1.4.3. VSwitch management

21.1.4.3.1. Create a vSwitch

A vSwitch is a basic network component in a virtual private cloud (VPC) and is used to connect cloud resources.

Context

After you create a VPC, you can create vSwitches to divide the VPC into one or more subnets. vSwitches within the same VPC can communicate with each other. Cloud resources must be deployed within the CIDR blocks of vSwitches. You can deploy applications in zones that are managed by different vSwitches to improve service availability.

> **Note** vSwitches do not support multicasting or broadcasting.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **vSwitches**.
3. Select the region of the VPC in which you want to create a vSwitch.
4. On the **vSwitches** page, click **Create vSwitch**.
5. On the **vSwitch** page, configure the vSwitch and click **OK**. The following table describes the parameters.

Parameter	Description
Organization	Select the organization to which the vSwitch belongs.
Resource Set	Select the resource set to which the vSwitch belongs.
Region	Select the region where you want to deploy the vSwitch.
Zone	Select the zone to which the vSwitch belongs. In a VPC, a vSwitch can belong to only one zone. However, you can deploy cloud resources in vSwitches that reside in different zones to achieve cross-zone disaster recovery. **Note** A cloud instance can be deployed in only one vSwitch.
Sharing Scope	Select the participants that can use the vSwitch to create resources. ○ **Current Resource Set**: Only the administrator of the current resource set can use the vSwitch to create resources. ○ **Current Organization and Subordinate Organization**: Only the administrators of the current organization and its subordinate organization can use the vSwitch to create resources. ○ **Current Organization**: Only the administrator of the current organization can use the vSwitch to create resources.
VPC	Select the VPC for which you want to create the vSwitch.
Dedicated for Out-of-cloud Physical Machines	Specify whether the vSwitch to be created is dedicated for bare-metal servers. For more information about bare-metal servers, see the **VPC bare-metal server features** topic in *BMS user guide*.

Parameter	Description
vSwitch Name	Enter a name for the vSwitch. The name must be 2 to 128 characters in length and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.
IPv4 CIDR Block	Enter an IPv4 CIDR block for the vSwitch. ○ You must specify the IP address range for the vSwitch in CIDR form. The CIDR block size for a vSwitch is between a 16-bit mask and a 29-bit mask. It means that 8 to 65,536 IP addresses can be provided. ○ The CIDR block of a vSwitch must be a subset of the CIDR block of the VPC network to which the vSwitch belongs. ○ The first and last three IP addresses of a vSwitch are reserved. For example, if the CIDR block of a vSwitch is 192.168.1.0/24, the IP addresses 192.168.1.0, 192.168.1.253, 192.168.1.254, and 192.168.1.255 are reserved. ○ The CIDR block of a vSwitch cannot be the same as the destination CIDR block in a route entry of the VPC to which the vSwitch belongs. However, the CIDR block of the vSwitch can be a subset of the destination CIDR block of the route entry. ○ After a vSwitch is created, you cannot modify its CIDR block.
IPv6 CIDR Block	Specify an IPv6 CIDR block for the vSwitch. ○ You must check whether IPv6 is enabled for the specified VPC. If not, you cannot assign an IPv6 CIDR block to the vSwitch. ○ If yes, you can enter a decimal number ranging from 0 to 255 to define the last 8 bits of the IPv6 CIDR block of the vSwitch. For example, if the IPv6 CIDR block of the VPC that contains the vSwitch is 2xx1:db8::/64, you can enter 255 (FF in the hexadecimal system) in this field to define the IPv6 CIDR block of the vSwitch as 2xx1:db8:ff::/64.
Description	Enter a description for the vSwitch. The description must be 2 to 256 characters in length and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.

21.1.4.3.2. Create cloud resources in a vSwitch

You cannot directly deploy cloud resources in a virtual private cloud (VPC). You can deploy cloud resources only in a vSwitch that belongs to a VPC. This topic describes how to create cloud resources in a vSwitch.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **vSwitches**.
3. Select the region of the VPC to which the vSwitch belongs.
4. On the **vSwitches** page, find the vSwitch, click **Create** in the **Actions** column, and select the cloud resource that you want to create.

   You can create Elastic Compute Service (ECS) instances, ApsaraDB RDS instances, and Server Load Balancer (SLB) instances in a vSwitch.
5. On the page that appears, set the parameters.

21.1.4.3.3. Modify the name and description of a VSwitch

This topic describes how to modify the name and description of a VSwitch.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **vSwitches**.
3. Select the region of the VPC in which you want to create a vSwitch.
4. On the **VSwitches** page, find the target VSwitch and click **Manage** in the **Actions** column.
5. In the **VSwitch Basic Information** section, click **Edit** next to **Name**. In the dialog box that appears, enter a new name for the VSwitch and click **OK**.

   The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), and hyphens (-). It must start with a letter or a Chinese character.
6. Click **Edit** next to **Description**. In the dialog box that appears, enter a new description for the VSwitch and click OK.

   The description must be 2 to 256 characters in length and cannot start with `http://` or `https://`.

21.1.4.3.4. Delete a vSwitch

This topic describes how to delete a vSwitch. After you delete a vSwitch, you cannot deploy cloud resources in it.

Prerequisites

Before you delete a vSwitch, make sure that the following conditions are met:

- You have deleted all the resources deployed in the vSwitch, such as Elastic Compute Service (ECS) instances, Server Load Balancer (SLB) instances, and ApsaraDB RDS instances.
- You have deleted all the resources associated with the vSwitch, such as high-availability virtual IP addresses (HAVIPs) and Source Network Address Translation (SNAT) entries.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **vSwitches**.
3. Select the region of the VPC in which you want to create a vSwitch.
4. On the **vSwitch** page, find the vSwitch and click **Delete** in the **Actions** column.
5. In the **Delete vSwitch** dialog box, click **OK**.

21.1.5. Route tables

21.1.5.1. Overview

Routes and route tables

Each virtual private cloud (VPC) comes with a default route table pre-configured with system route entries that direct traffic flowing in and out of the VPC. You cannot create or delete system route entries. However, you can create custom route entries to route traffic destined for specific Classless Inter-domain Routing (CIDR) blocks to the destinations that you define.

Route tables

When you create a VPC, it automatically has a system route table that controls the routing for all VSwitches (subnets) within the VPC by default. You cannot create or delete the system route table of a VPC.

Each *route entry* in the route table defines a route that is used to direct traffic, and consists of multiple fields, including the destination CIDR block, the next hop for the traffic, and the type of the next hop. Route entries are classified into system route entries and custom route entries.

System routes

After you create a VPC, the system automatically adds the following system routes to the route table:

- A route entry with a destination CIDR block of 100.64.0.0/10. This route is used for communication among cloud resources within the VPC.
- Route entries with destination CIDR blocks same as the CIDR blocks of the VSwitches in this VPC. Such routes are used for communication among cloud resources within VSwitches.

For example, if you create a VPC and specify 192.168.0.0/16 as its CIDR block, and then create two VSwitches whose CIDR blocks are 192.168.1.0/24 and 192.168.0.0/24, three system routes listed in the following table are automatically added to the route table of the VPC.

Destination CIDR block	Next hop	Route entry type
100.64.0.0/10	-	System route
192.168.1.0/24	-	System route
192.168.0.0/24	-	System route

Custom routes

You can add custom routes to replace system routes or route traffic to specified destinations. You can specify the following next hop types when you create a custom route:

- ECS instance: Traffic destined for the destination CIDR block is forwarded to a specified Elastic Compute Service (ECS) instance in the VPC.

  You can select this type if you want to access the Internet or other applications through the applications deployed on an ECS instance.
- VPN gateway: Traffic destined for the destination CIDR block is forwarded to a specified VPN gateway.

  You can select this type if you want to connect a VPC to another VPC or a local network through a VPN connection.
- NAT gateway: Traffic destined for the destination CIDR block is forwarded to a specified NAT gateway.

  You can select this type if you want to connect a VPC to the Internet by using a NAT gateway.
- Router interface (to VPC): Traffic destined for the destination CIDR block is forwarded to a specified VPC.

  You can select this type if you want to connect two VPCs by using Express Connect.
- Router interface (to VBR): Traffic destined for the destination CIDR block is forwarded to a specified Virtual Border Router (VBR).

  You can select this type if you want to connect a VPC to an on-premises network by using Express Connect.
- Secondary ENI: Traffic destined for the destination CIDR block is forwarded to a specified secondary Elastic Network Interface (ENI).

IPv6 routes

If IPv6 is enabled for your VPC, the following route entries are automatically added to the system route table of the VPC:

- A custom route entry with a destination CIDR block of ::/0 and whose next hop is the IPv6 gateway. This route is used to direct traffic between the cloud resources deployed in the VPC and the Internet by using IPv6 addresses.

- System route entries whose destination CIDR blocks are the IPv6 CIDR blocks of the VSwitches in the VPC. Such routes are used for communication within the VSwitches.

Routing rules

If one destination address matches more than one route entry in a route table, the system selects an entry by implementing the longest prefix match algorithm, whereby the most specific of the matching entries, the one with the longest subnet mask, is used to route traffic.

The following table describes a route table of a VPC.

Destination CIDR block	Next hop type	Next hop	Route entry type
100.64.0.0/10	-	-	System route
192.168.0.0/24	-	-	System route
0.0.0.0/0	Instance	i-12345678	Custom route
10.0.0.0/24	Instance	i-87654321	Custom route

As shown in the preceding table, the route entries destined for `100.64.0.0/10` and `192.168.0.0/24` are system route entries, and the two destined for `0.0.0.0/0` and `10.0.0.0/24` are custom route entries. Traffic destined for `0.0.0.0/0` is forwarded to the ECS instance `i-12345678`, and traffic destined for `10.0.0.0/24` is forwarded to the ECS instance `i-87654321`. According to the longest prefix match algorithm, traffic destined for `10.0.0.1` is forwarded to the ECS instance `i-87654321`, and traffic destined for `10.0.1.1` is forwarded to the ECS instance `i-12345678`.

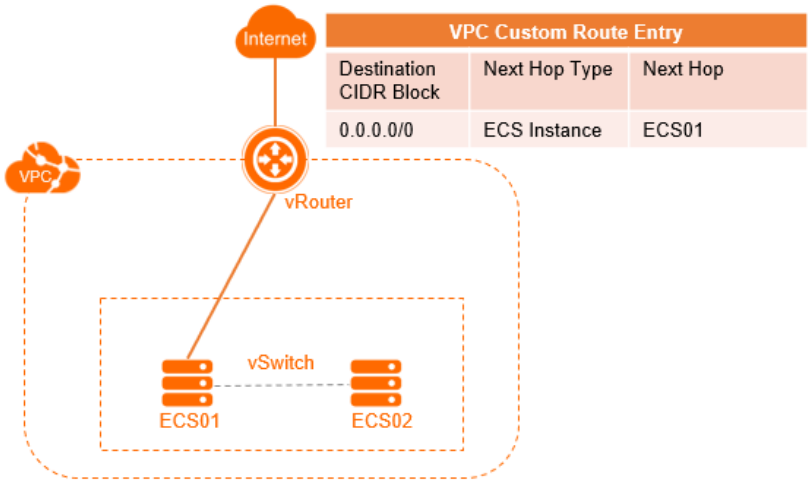
Routing examples

You can add custom route entries to a route table to control inbound and outbound traffic for a VPC.

- Routing within a VPC

  As shown in the following figure, a NAT gateway is deployed on an ECS instance (ECS01) in a VPC. To enable the cloud resources in this VPC to access the Internet by this ECS instance, you can add the following route entry to the route table.

Destination CIDR block	Next hop type	Next hop
0.0.0.0/0	ECS instance	ECS01



- Connect two VPCs by using Express Connect

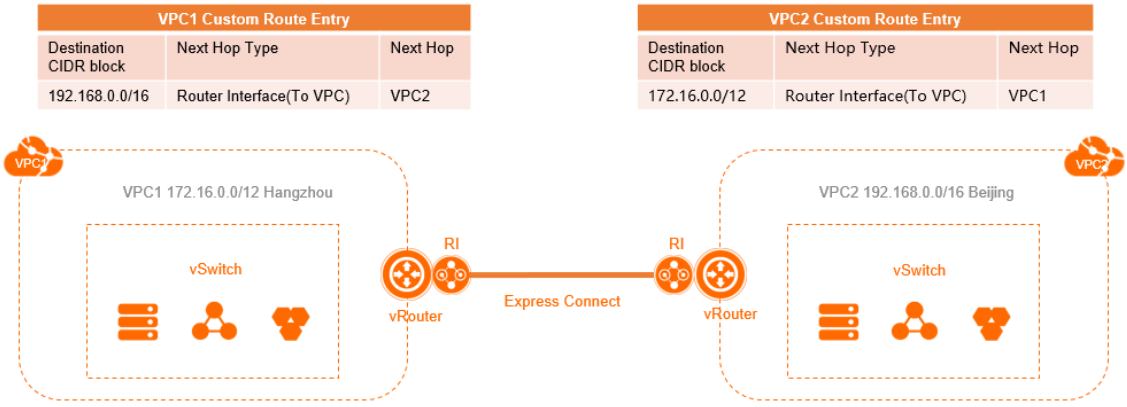
  As shown in the following figure, VPC1 (172.16.0.0/12) needs to be connected to VPC2 (192.168.0.0/16) by using Express Connect. After you create router interfaces for interconnection, you must add the following route entries in the route table of each VPC respectively.

  - VPC1

Destination CIDR block	Next hop type	Next hop
192.168.0.0/16	Router interface (to VPC)	VPC2

  - VPC2

Destination CIDR block	Next hop type	Next hop
172.16.0.0/12	Router interface (to VPC)	VPC1



- Connect two VPCs by using VPN gateways

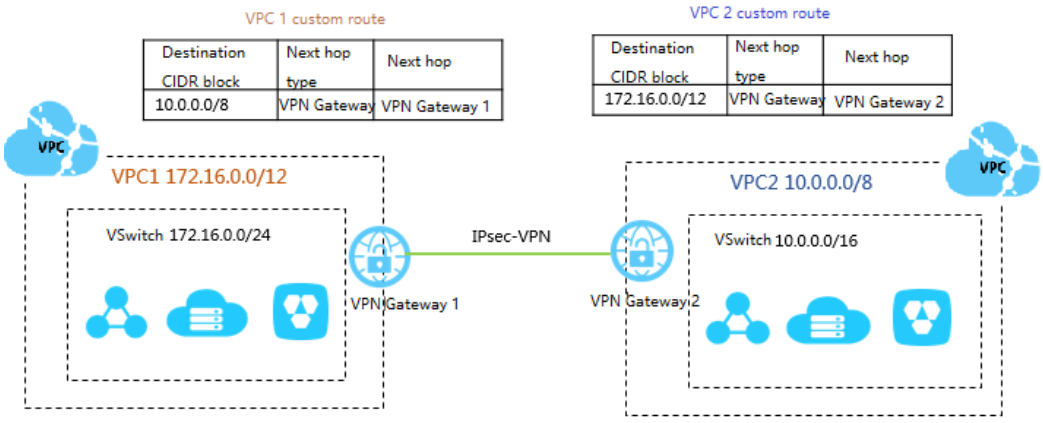
  As shown in the following figure, VPC1 (172.16.0.0/12) needs to be connected to VPC2 (10.0.0.0/8) with VPN gateways. After you configure the VPN gateways, you must add the following route entries in the route table of each VPC respectively.

- VPC1

Destination CIDR block	Next hop type	Next hop
10.0.0.0/8	VPN gateway	VPN gateway 1

- VPC2

Destination CIDR block	Next hop type	Next hop
172.16.0.0/12	VPN gateway	VPN gateway 2



- Connect a VPC to an on-premises data center by using Express Connect

  As shown in the following figure, a VPC needs to be connected to an on-premises data center by using Express Connect. After you configure an Express Connect circuit and a VBR, you must add the following route entries for related networks and devices:

  - VPC

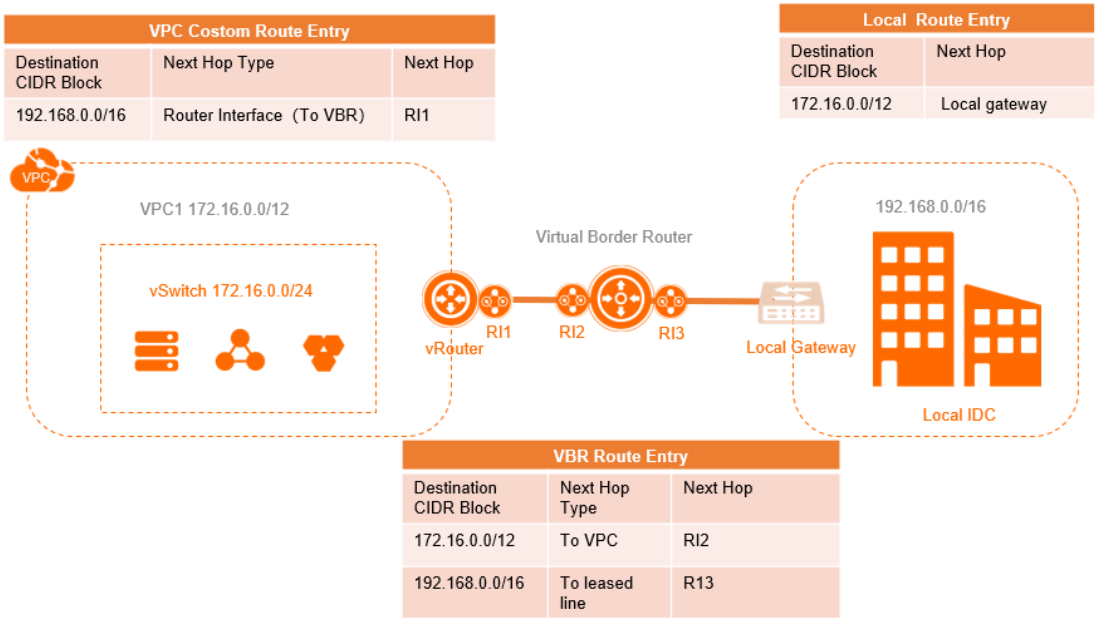
Destination CIDR block	Next hop type	Next hop
192.168.0.0/16	Router interface (general routing)	Router interface RI1

  - VBR

Destination CIDR block	Next hop type	Next hop
192.168.0.0/16	To the Express Connect circuit	Router interface RI3
172.16.0.0/12	To VPC	Router interface RI2

- On-premises network

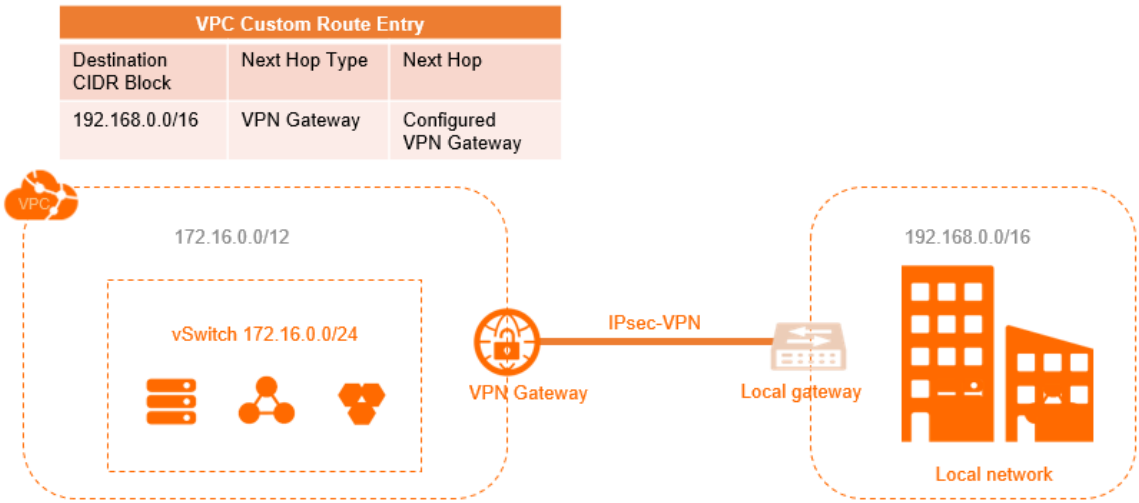
Destination CIDR block	Next hop type	Next hop
172.16.0.0/12	-	A specified local gateway device



- Connect a VPC to an on-premises data center by using VPN gateways

  As shown in the following figure, a VPC (172.16.0.0/12) needs to be connected to an on-premises data center (192.168.0.0/16). After you configure the VPN gateway, you must add the following route entry to the route table of the VPC.

Destination CIDR block	Next hop type	Next hop
192.168.0.0/16	VPN gateway	A specified VPN gateway



21.1.5.2. Add a custom route entry

This topic describes how to add a custom route entry. After you create a virtual private cloud (VPC), the system creates a default route table and adds system route entries to the route table for traffic management. You cannot create or delete system route entries. However, you can create custom route entries to route traffic from source CIDR blocks to specific destinations.

Context

Each item in the route table is a route entry. A route entry, which specifies the destination for network traffic, consists of the destination CIDR block, next hop type, and next hop. Route entries are classified into system route entries and custom route entries.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **Route Tables**.
3. In the top navigation bar, select the region to which the route table belongs.
4. On the **Route Tables** page, find the route table and click **Manage** in the **Actions** column.
5. In the **Route Table Details** section, click **Add Route Entry**.
6. In the **Add Route Entry** panel, set the following parameters and click **OK**.

Parameter	Description
Name	Enter a name for the route entry. The name must be 2 to 128 characters and can contain digits, underscores (_), and hyphens (-). The name must start with a letter.
Destination CIDR Block	Enter a destination CIDR block for the route entry.

Parameter	Description
Next Hop Type	Select the next hop type. Valid values: ○ **ECS Instance**: Traffic destined for the specified CIDR block is routed to the specified Elastic Compute Service (ECS) instance. Select this type if you want to route traffic to an ECS instance for centralized traffic forwarding and management. For example, when an ECS instance is configured as the public-facing gateway to manage the traffic from other ECS instances to the Internet. ○ **HaVip Address**: Traffic destined for the specified CIDR block is routed to the specified high-availability virtual IP address (HAVIP). ○ **VPN Gateway**: Traffic destined for the specified CIDR block is routed to the specified VPN gateway. ○ **NAT Gateway**: Traffic destined for the specified CIDR block is routed to the specified NAT gateway. ○ **Secondary ENI**: Traffic destined for the specified CIDR block is routed to the specified secondary elastic network interface (ENI). ○ **Router Interface (To VPC)**: Traffic destined for the specified CIDR block is routed to the specified VPC. Select this type if you want to connect VPCs through Express Connect. ○ **Router Interface (To VBR)**: Traffic destined for the specified CIDR block is routed to the router interface that is associated with a virtual border router (VBR). Select this type if you want to connect the VPC to a data center through Express Connect. If you select Router Interface (To VBR), you must also select a routing mode: ■ **General Routing**: Select an associated router interface. ■ **Active/Standby Routing**: Select two instances as the next hop. The active route has a weight of 100 and the standby route has a weight of 0. The standby route takes over when the active route fails the health check. ■ **Load Balancing**: Select two to four router interfaces as the next hop. The peer router of each router interface must be a VBR. Valid values of the instance weight: 1 to 255. The value must be an integer and the default value is 100. The weights of the selected instances must be the same. This way, traffic can be evenly distributed to the next-hop instances.
ECS Instance/VPN Gateway/NAT Gateway/Secondary ENI/HAVIP/VPC	Select the next hop instance.

21.1.5.3. Export route entries

This topic describes how to export route entries from a route table for backup.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **Route Tables**.
3. In the top navigation bar, select the region to which the route table belongs.
4. On the **Route Tables** page, find the route table and click **Manage** in the **Actions** column.

5. In the **Route Table Details** section, click the **Route Entry List** tab, and then click **Export**.

   The route entries are exported to a *.csv* file in your local computer.

21.1.5.4. Modify a route table

This topic describes how to modify the name and description of a route table.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **Route Tables**.
3. In the top navigation bar, select the region to which the route table belongs.
4. On the **Route Tables** page, find the route table and click its ID.
5. In the **Route Table Details** section, click **Edit** next to **Name**. In the dialog box that appears, enter a new name for the route table and click **OK**.

   The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), and hyphens (-). It must start with a letter or a Chinese character.
6. Click **Edit** next to **Description**. In the dialog box that appears, enter a new description of the route table, and click **OK**.

   The description must be 2 to 256 characters in length and cannot start with `http://` or `https://`.

21.1.5.5. Delete a custom route entry

This topic describes how to delete a custom route entry. A route table consists of one or more route entries that determine which way to forward traffic. Note that system route entries cannot be deleted.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **Route Tables**.
3. In the top navigation bar, select the region to which the route table belongs.
4. On the **Route Tables** page, find the route table and click its ID.
5. On the **Route Entry List** tab, find the target route entry and then click **Delete** in the **Actions** column.
6. In the **Delete Route Entry** dialog box, click **OK**.

21.1.6. HAVIPs

21.1.6.1. Overview

High-availability virtual IP addresses (HAVIPs) are private IP addresses that can be created and released as independent resources. You can use HAVIPs with high-availability (HA) software such as Keepalived to provide active/standby services. This improves the availability of your services.

Features

Each Elastic Compute Service (ECS) instance is assigned a private IP address as the primary IP address. If you want the ECS instance to use more than one private IP address, you can associate HAVIPs with the ECS instance. Both the primary IP address and HAVIPs of an ECS instance can be used to access networks. In addition, you can use HAVIPs with HA software such as Keepalived to provide active/standby services. This improves the availability of your services. You can associate an HAVIP with ECS instances in the following ways:

- Directly associate an HAVIP with ECS instances.

Each HAVIP can be associated with two ECS instances. After an HAVIP is associated with ECS instances, the ECS instances can send Address Resolution Protocol (ARP) messages to advertise the HAVIP. After the ECS instances advertise the HAVIP, one of the ECS instances serves as the primary instance, and the other ECS instance serves as the secondary instance. If the primary ECS instance is down, the secondary ECS instance takes over to provide services.

- Attach a secondary elastic network interface (ENI) to each ECS instance. Then, associate an HAVIP with the secondary ENIs.

  Each HAVIP can be associated with two ECS instances. After an HAVIP is associated with ECS instances, the ECS instances can send ARP messages to advertise the HAVIP. After the ECS instances advertise the HAVIP, one of the ECS instances serves as the primary instance, and the other ECS instance serves as the secondary instance. If the primary ECS instance is down, the secondary ECS instance takes over to provide services.

  > **Note** Before you associate an HAVIP with secondary ENIs, make sure that the secondary ENIs are attached to different ECS instances.

HAVIPs have the following features:

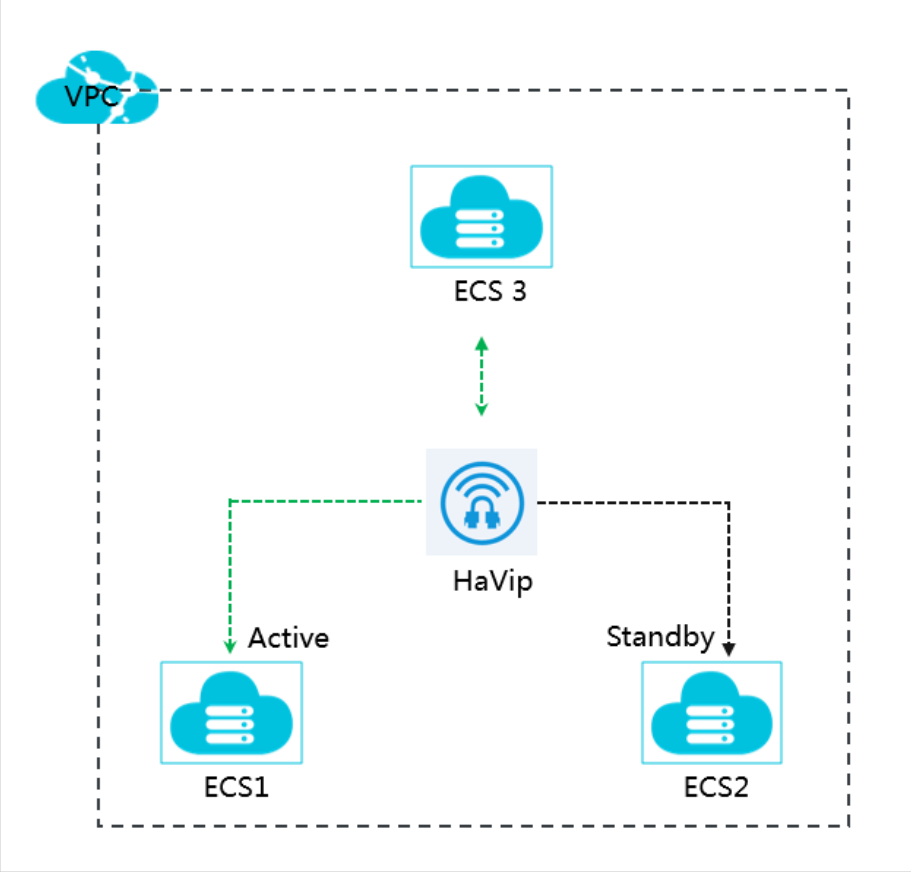
- HAVIPs are floating private IP addresses and are not statically assigned to ECS instances. HAVIPs can be associated with or disassociated from ECS instances through ARP announcements.
- You must associate HAVIPs with ECS instances or secondary ENIs within the same vSwitch. This means that the ECS instances or secondary ENIs must belong to the same subnet.
- You can associate each HAVIP with two ECS instances or two secondary ENIs. However, you cannot associate an HAVIP with an ECS instance and a secondary ENI.

Scenarios

HAVIPs can be used in different ways to meet the requirements of diverse scenarios:

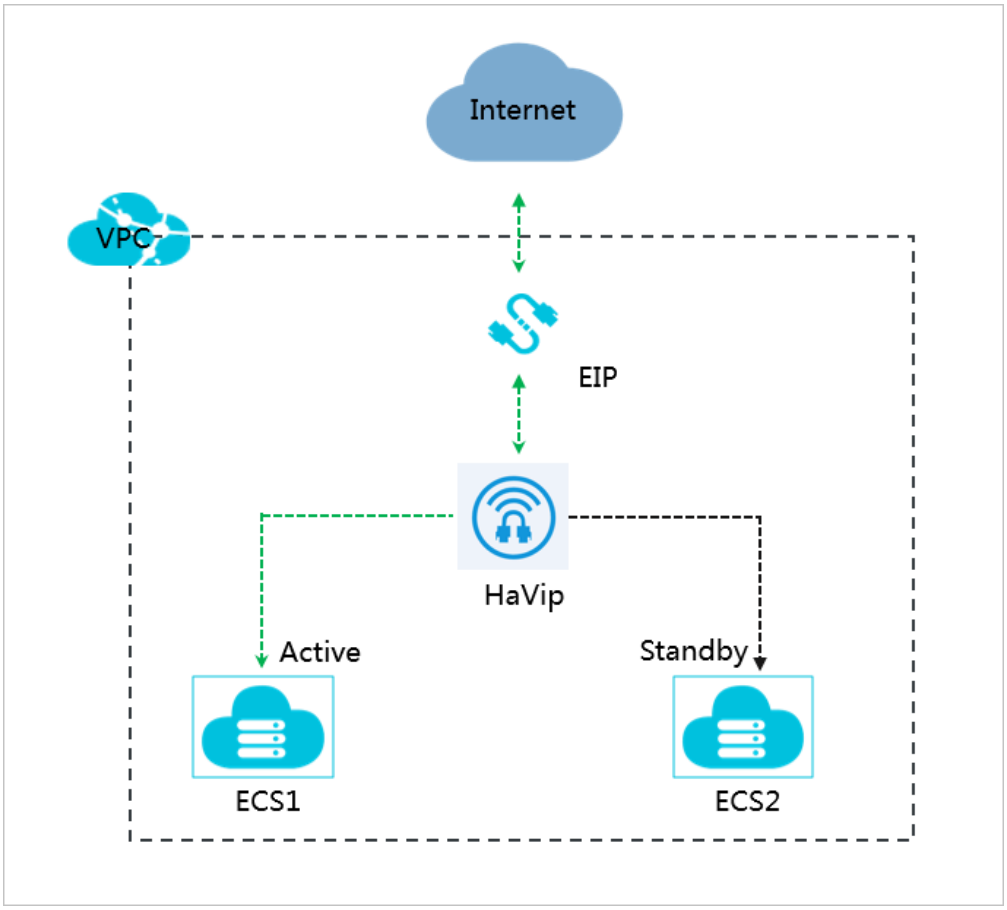
- Scenario 1: Private network-facing HA services

  In the following figure, two ECS instances are assigned the same HAVIP. Keepalived is configured for the ECS instances to provide a private network-facing HA service. Other instances in the same virtual private cloud (VPC) can access this service over the private network. The HAVIP functions as the service address. If the primary ECS instance is down, the secondary ECS instance takes over. This improves the availability of your services.



- Scenario 2: Internet-facing HA services

  In the following figure, two ECS instances are assigned the same HAVIP. Keepalived is configured and the HAVIP is associated with an elastic IP address (EIP) for the ECS instances to provide an Internet-facing HA service. The EIP that is associated with the HAVIP functions as the service address. If the primary ECS instance is down, the secondary ECS instance takes over. This improves the availability of your services.



Limits

Before you use HAVIPs, take note of the following limits:

Item	Default limit
Number of HAVIPs that can be created in each VPC	5
Number of HAVIPs that can be associated with each ECS instance	5
Number of HAVIPs that can be associated with each ENI	5
Number of ECS instances that can be associated with each HAVIP	2
Number of ENIs that can be associated with each HAVIP	2
Number of route entries destined for an HAVIP in each VPC	5

Item	Default limit
Whether HAVIPs support broadcast or multicast communication	Not supported **Note** HAVIPs support only unicast. To implement high availability through third-party software such keepalived, you must modify the configuration file to change the communication method to unicast.

21.1.6.2. Create HAVIPs

High-availability virtual IP addresses (HAVIPs) are private IP addresses that can be created and released as independent resources. This topic describes how to create HAVIPs in the console.

Prerequisites

A VPC and vSwitches are created. For more information, see Create a VPC and Work with vSwitches.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **HaVip**.
3. In the top menu bar, select the region where you want to create the HAVIP.
4. On the **HaVip** page, click **Create HaVip**.
5. On the **Create a high-availability virtual IP address** page, set the following parameters and click **Submitted** to create an HAVIP.

Parameter	Description
Tissue	Select the organization to which the HAVIP belongs.
Resource Set	Select the resource set to which the HAVIP belongs.
Region	Select the region where you want to create the HAVIP.
Proprietary Network vpc	Select the VPC to which the HAVIP that you want to create belongs.
vswitch	Select the vSwitch to which the HAVIP that you want to create belongs.
Private IP Address	Specify a private IP address for the HAVIP. **Note** You must specify an idle private IP address that falls within the CIDR block of the vSwitch.

21.1.6.3. Associate HAVIPs with backend cloud resources

21.1.6.3.1. Associate HAVIPs with ECS instances

This topic describes how to associate high-availability virtual IP addresses (HAVIPs) with Elastic Compute Service (ECS) instances that are deployed in virtual private clouds (VPCs). After you associate an HAVIP with an ECS instance, the ECS instance can send Address Resolution Protocol (ARP) messages to advertise the HAVIP. This way, the ECS instance can use more than one private IP address. Each HAVIP can be associated with at most two ECS instances.

Prerequisites

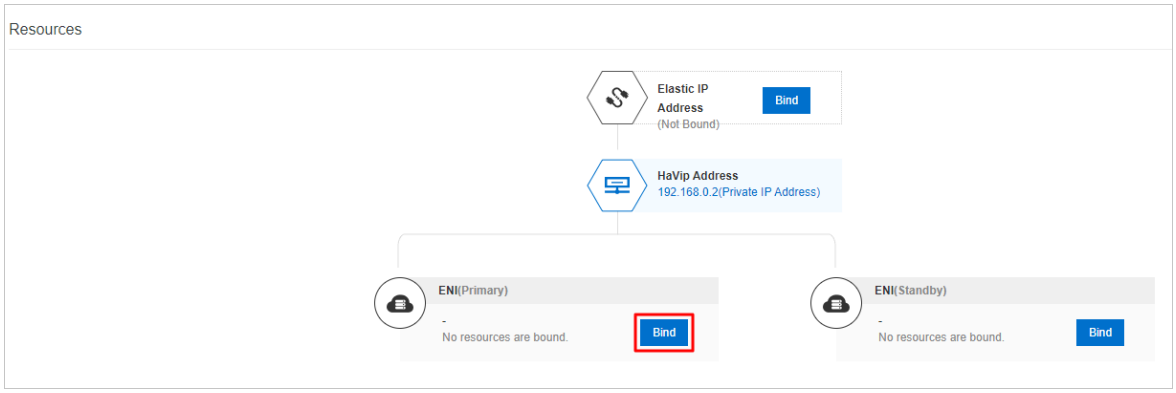
An ECS instance is created. For more information, see in the **Create an ECS instance** topic in **Quick Start** of *Elastic Compute Service User Guide*.

Context

You can associate an HAVIP with two ECS instances or two secondary ENIs. However, you cannot associate an HAVIP with an ECS instance and a secondary ENI.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **HaVip**.
3. In the top menu bar, select the region where the HAVIP is created.
4. On the **HaVip Addresses** page, find the HAVIP that you want to manage and click **Manage** in the **Actions** column.
5. In the **Resources** section, find **ENI (Primary)** or **ENI (Standby)** and click **Bind**.



6. In the dialog box that appears, set the following parameters to associate the HAVIP with an ECS instance.

Parameter	Description
Resource Type	Select the type of resource with which you want to associate the HAVIP. Supported resource types are: ○ **ECS Instance** ○ **Secondary ENI** In this example, **ECS Instance** is selected.
Bind Resource	Select the ECS instance with which you want to associate the HAVIP. The ECS instance that you select must meet the following requirements: ○ The ECS instance is deployed in a VPC. ○ The ECS instance and the HAVIP belong to the same vSwitch.

7. Click **OK**.

21.1.6.3.2. Associate an HAVIP with a secondary ENI

This topic describes how to associate a high-availability virtual IP address (HAVIP) with secondary ENIs that are attached to Elastic Compute Service (ECS) instances. Then, the ECS instances can send Address Resolution Protocol (ARP) messages to advertise the HAVIP. This way, the ECS instances can use more than one private IP address.

Prerequisites

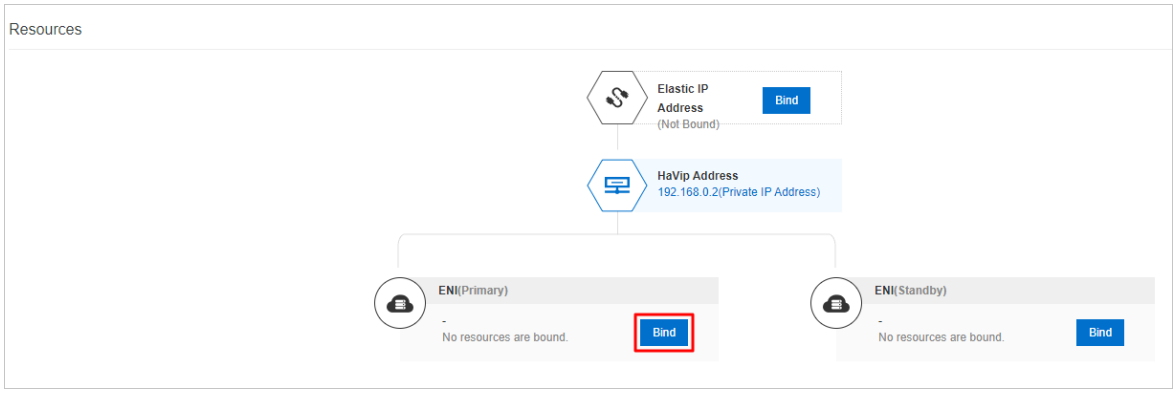
Secondary ENIs are created. For more information, see the **Create an Elastic Network Interface** topic in the **Elastic Network Interface** chapter of *Elastic Compute Service User Guide*.

Context

You can associate each HAVIP with two ECS instances or two secondary ENIs. However, you cannot associate an HAVIP with an ECS instance and a secondary ENI.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **HaVip**.
3. In the top navigation bar, select the region where the HAVIP is created.
4. On the **HaVip Details** page, find the HAVIP that you want to manage and click **Manage** in the **Actions** column.
5. In the **Resources** section, find **ENI (Primary)** or **ENI (Standby)** and click **Bind**.



6. In the dialog box that appears, set the following parameters to associate an HAVIP with a secondary ENI.

Parameter	Description
Resource Type	Select the type of resource with which you want to associate the HAVIP. Supported types of resources are: ○ **ECS Instances** ○ **Secondary ENI** In this example, **Secondary ENI** is selected.
Bind Resource	Select the secondary ENI with which you want to associate the HAVIP. The secondary ENI and the HAVIP must belong to the same vSwitch.

7. Click **OK**.

21.1.6.4. Associate HAVIPs with EIPs

This topic describes how to associate high-availability virtual IP addresses (HAVIPs) with elastic IP addresses (EIPs). After you associate an HAVIP with an EIP, the HAVIP can use the EIP to provide services over the Internet.

Prerequisites

An EIP is created. For more information, see **Create an EIP** in **Quick Start** of the *Elastic IP Address User Guide*.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **HaVip**.
3. In the top menu bar, select the region where the HAVIP is created.
4. On the **HaVip Addresses** page, find the HAVIP that you want to manage, and choose **More > Bind EIP Address** in the **Actions** column.
5. In the dialog box that appears, select the EIP with which you want to associate the HAVIP and click **OK**.

   The EIP with which you want to associate the HAVIP must meet the following requirements:

   - The EIP and HAVIP are created in the same region.
   - The EIP must be in the Available state.

21.1.6.5. Disassociate HAVIPs from backend cloud resources

21.1.6.5.1. Disassociate HAVIPs from ECS instances

This topic describes how to disassociate high-availability virtual private IP addresses (HAVIPs) from Elastic Compute Service (ECS) instances. After you disassociate an HAVIP from an ECS instance, the ECS instance cannot send Address Resolution Protocol (ARP) messages to advertise the HAVIP.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **HaVip**.
3. In the top menu bar, select the region where the HAVIP is created.
4. On the **HaVip** page, find the HAVIP that you want to manage and click **Manage** in the **Actions** column.
5. In the **Resources** section, find the ECS instance that you want to manage and click **Unbind**.
6. In the message that appears, click **OK**.

21.1.6.5.2. Disassociate HAVIPs from secondary ENIs

This topic describes how to disassociate high-availability IP addresses (HAVIPs) from secondary elastic network interfaces (ENIs). After you disassociate an HAVIP from a secondary ENI, the Elastic Compute Service (ECS) instance with which the secondary ENI is attached cannot send Address Resolution Protocol (ARP) messages to advertise the HAVIP.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **HaVip**.
3. In the top menu bar, select the region where the HAVIP is created.
4. On the **HaVip** page, find the HAVIP that you want to manage and click **Manage** in the **Actions** column.

5. In the **Resources** section, find the secondary ENI that you want to manage and click **Unbind**.
6. In the message that appears, click **OK**.

21.1.6.6. Disassociate an EIP from an HAVIP

This topic describes how to disassociate a high-availability virtual IP address (HAVIP) from an elastic IP address (EIP). After the disassociation, you can no longer use the HAVIP to provide your instances with Internet access.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **HaVip Addresses**.
3. Select the region of the HAVIP.
4. On the **HaVip Addresses** page, find the target HAVIP and then choose **More > Unbind with EIP** in the **Actions** column.
5. In the dialog box that appears, click **OK**.

21.1.6.7. Delete HAVIPs

This topic describes how to delete a high-availability virtual IP address (HAVIP) that you no longer need.

Prerequisites

- The HAVIP that you want to delete is not associated with an elastic IP address (EIP). If the HAVIP that you want to delete is associated with an EIP, disassociate the HAVIP from the EIP first. For more information, see Disassociate HAVIPs from EIPs.
- The HAVIP that you want to delete is not associated with an Elastic Compute Service (ECS) instance. If the HAVIP that you want to delete is associated with an ECS instance, disassociate the HAVIP from the ECS instance. For more information, see Disassociate an HAVIP from an ECS instance.
- The HAVIP that you want to delete is not associated with a secondary elastic network interface (ENI). If the HAVIP that you want to delete is associated with a secondary ENI, disassociate the HAVIP from the secondary ENI first. For more information, see Disassociate HAVIPs from secondary ENIs.

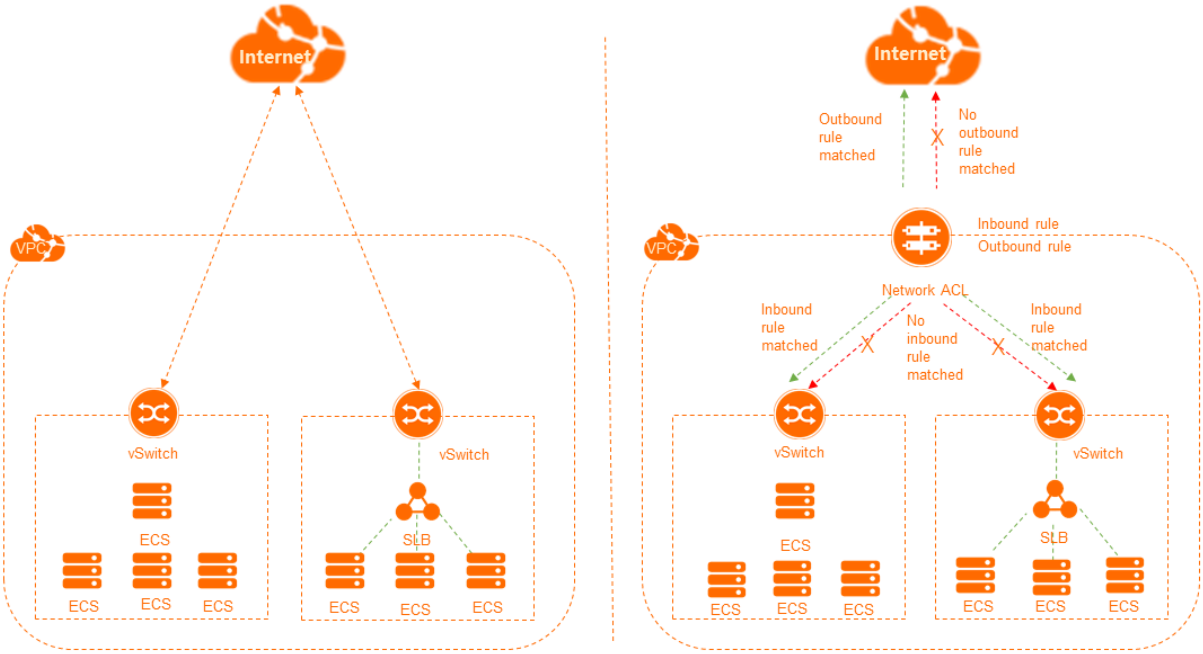
Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **HaVip**.
3. In the top navigation bar, select the region where the HAVIP is created.
4. On the **HaVip** page, find the HAVIP that you want to manage and choose **More > Delete** in the **Actions** column.
5. In the dialog box that appears, click **OK**.

21.1.7. Network ACLs

21.1.7.1. Overview

Network access control lists (ACLs) provided by Virtual Private Cloud (VPC) allow you to manage network access permissions. You can create network ACL rules and associate a network ACL with a vSwitch. This allows you to control inbound and outbound traffic of Elastic Compute Service (ECS) instances that are associated with the vSwitch.



Features

Network ACLs have the following features:

- A network ACL is used to filter inbound and outbound network traffic of ECS instances that are associated with a vSwitch in a VPC. The network traffic forwarded to ECS instances by Server Load Balancer (SLB) instances is also filtered.
- Network ACLs are stateless. You must set both inbound and outbound rules. Otherwise, the system may fail to respond to requests.
- If you create a network ACL that does not contain any rule, all inbound and outbound access are rejected.
- If a network ACL is associated with a vSwitch, the network ACL does not filter the traffic forwarded between ECS instances that are associated with the vSwitch.

Rule descriptions

You can add rules to or delete rules from a network ACL. Changes to the rules are automatically synchronized to the associated vSwitch. By default, an inbound and outbound rule are automatically added to a newly created network ACL. These rules allow all inbound and outbound network traffic transmitted through the associated vSwitch. You can delete the default rules. The following table lists the default inbound and outbound rules.

- Default inbound rule

Priority	Protocol type	Source CIDR block	Destination port range	Action	Type
1	all	0.0.0.0/0	-1/-1	Allow	Custom

- Default outbound rule

Priority	Protocol type	Destination CIDR block	Destination port range	Action	Type
1	all	0.0.0.0/0	-1/-1	Allow	Custom

A network ACL contains the following parameters:

- Priority: A smaller value indicates a higher priority. The system compares traffic requests with rules in

descending order of priority starting from the rule whose priority is 1. If a request meets a rule, the system applies the rule to the request and ignores the other rules.

For example, the following rules are added and requests destined for IP address 172.16.0.1 are sent from an ECS instance. In the following table, the requests meet Rules 2 and 3. Rule 2 has a higher priority than Rule 3. Therefore, the system applies Rule 2. Based on the action of Rule 2, the requests are denied.

Priority	Protocol type	Destination CIDR block	Destination port range	Action	Type
1	all	10.0.0.0/8	-1/-1	Allow	Custom
2	all	172.16.0.0/12	-1/-1	Deny	Custom
3	all	172.16.0.0/12	-1/-1	Allow	Custom

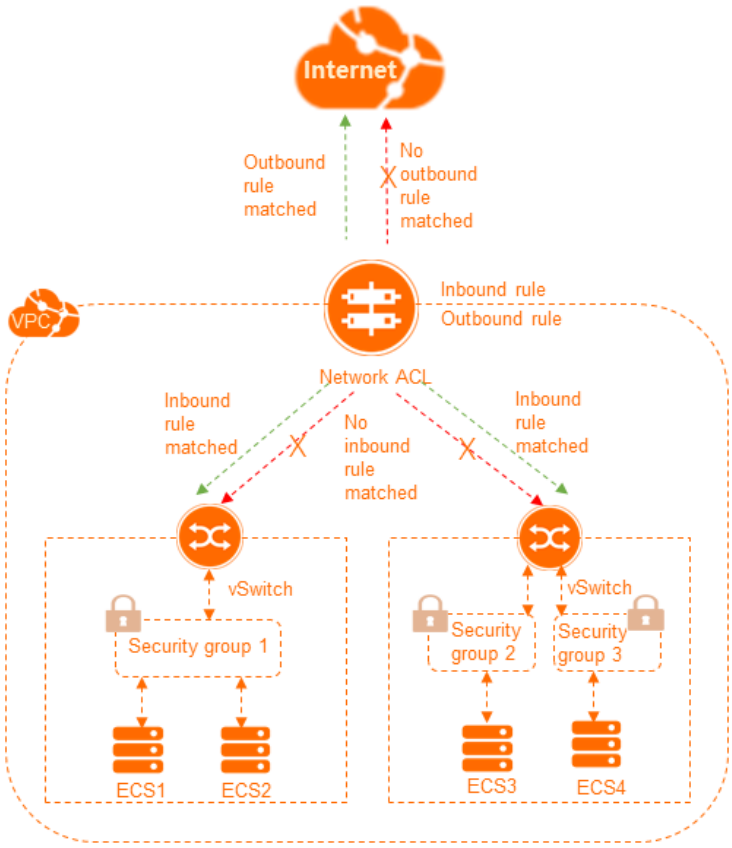
- Policy: indicates whether to allow or deny specific traffic.
- Protocol: the protocol type. Valid values: All, ICMP, GRE, TCP, and UDP.
- Source CIDR block: the source CIDR block from which inbound traffic is transmitted.
- Destination CIDR block: the destination CIDR block to which outbound traffic is transmitted.
- Destination port range: the range of destination ports to which the inbound rule applies.
- Destination port range: the range of destination ports to which the outbound rule applies.

Comparison between network ACLs and security groups

Network ACLs control data transmitted through associated vSwitches while security groups control data transmitted through associated ECS instances. The following table lists the differences between network ACLs and security groups.

Network ACL	Security group
Applied to vSwitches.	Applied to instances.
Stateless: Returned traffic must be allowed by rules.	Stateful: Returned traffic is automatically allowed and not affected by any rule.
The system compares traffic requests with rules in descending order of priority. Not all rules are compared.	All rules are compared before a rule is applied.
Each vSwitch can be associated with only one network ACL.	Each ECS instance can be added to more than one security group.

The following figure shows how network ACLs and security groups are applied to ensure network security.



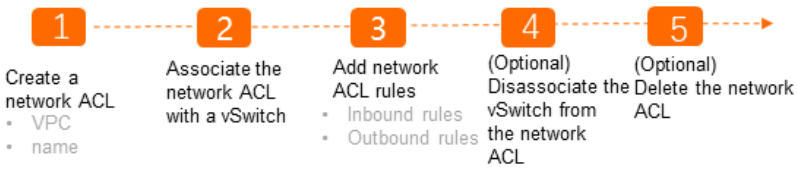
Limits

Before you use network ACLs, take note of the following limits.

Item	Default limit
Number of network ACLs that can be created in each VPC	200
Number of network ACLs that can be associated with each vSwitch	1
Number of rules that can be added to a network ACL	• Inbound rules: 20 • Outbound rules: 20

Procedure

The following flowchart shows how to use a network ACL.



21.1.7.2. Scenarios

If you are familiar with the ports that are commonly used by ECS instances, you can specify them in access control list (ACL) rules to facilitate precise network traffic filtering. This topic describes the ports that are commonly used by ECS instances and the application scenarios of these ports.

Ports

The following table lists the ports and the services that use these ports.

Port	Service	Description
21	FTP	The FTP port. It is used to upload and download files.
22	SSH	The SSH port. It is used to log on to Linux instances in the command line method by using username and password pairs.
23	Telnet	The Telnet port. It is used to remotely log on to ECS instances.
25	SMTP	The SMTP port. It is used to send emails.
80	HTTP	The HTTP port. It is used to access services such as IIS, Apache, and NGINX.
110	POP3	The POP3 port. It is used to send and receive emails.
143	IMAP	The Internet Message Access Protocol (IMAP) port. It is used to receive emails.
443	HTTPS	The HTTPS port. It is used to access services. The HTTPS protocol can implement encrypted and secure data transmission.
1433	SQL Server	The TCP port of SQL Server. It is used for SQL Server to provide external services.
1434	SQL Server	The UDP port of SQL Server. It is used to return the TCP/IP port occupied by SQL Server.
1521	Oracle	The Oracle communication port. ECS instances that run Oracle SQL must have this port open.
3306	MySQL	The MySQL port. It is used for MySQL databases to provide external services.
3389	Windows Server Remote Desktop Services	The Windows Server Remote Desktop Services port. It is used to log on to a Windows instance.
8080	Proxy port	An alternative to port 80. It is commonly used for WWW proxy services.

Custom network ACLs

Inbound rules and Outbound rules describe a network ACL example for VPCs that support IPv4 addresses only.

- The inbound rules in effective order 1, 2, 3, and 4 respectively allow HTTP, HTTPS, SSH, and RDP traffic to the vSwitch. Outbound response rules are those in effective order 3.
- The outbound rules in effective order 1 and 2 respectively allow HTTP and HTTPS traffic from the vSwitch. Outbound response rules are those in effective order 5.
- The inbound rule in effective order 6 denies all inbound IPv4 traffic. This rule ensures that packets that do not match any other rules are denied.
- The outbound rule in effective order 4 denies all outbound IPv4 traffic. This rule ensures that packets that do

not match any other rules are denied.

> **Note** An inbound or outbound rule must correspond to an inbound or outbound rule that allows response traffic.

Inbound rules

Effective order	Protocol	Source IP addresses	Destination port range	Action	Description
1	TCP	0.0.0.0/0	80/80	Accept	Allows inbound HTTP traffic from any IPv4 addresses.
2	TCP	0.0.0.0/0	443/443	Accept	Allows inbound HTTPS traffic from any IPv4 addresses.
3	TCP	0.0.0.0/0	22/22	Accept	Allows inbound SSH traffic from any IPv4 addresses.
4	TCP	0.0.0.0/0	3389/3389	Accept	Allows inbound RDP traffic from any IPv4 addresses.
5	TCP	0.0.0.0/0	32768/65535	Accept	Allows inbound IPv4 traffic from the Internet. This port range is for reference only. For more information on how to select appropriate ephemeral ports, see Ephemeral ports.
6	All	0.0.0.0/0	-1/-1	Drop	Denies all inbound IPv4 traffic.

Outbound rules

Effective order	Protocol	Destination IP addresses	Destination port range	Action	Description
1	TCP	0.0.0.0/0	80/80	Accept	Allows outbound IPv4 HTTP traffic from the vSwitch to the Internet.
2	TCP	0.0.0.0/0	443/443	Accept	Allows outbound IPv4 HTTPS traffic from the vSwitch to the Internet.
3	TCP	0.0.0.0/0	32768/65535	Accept	Allows outbound IPv4 traffic from the vSwitch to the Internet. This port range is for reference only. For more information on how to select appropriate ephemeral ports, see Ephemeral ports.
4	All	0.0.0.0/0	-1/-1	Drop	Denies all outbound IPv4 traffic.

Network ACLs for SLB

If the ECS instance in the vSwitch acts as the backend server of an SLB instance, you must add the following network ACL rules.

- Inbound rules

Effective order	Protocol	Source IP addresses	Destination port range	Action	Description
1	SLB listener protocol	Client IP addresses allowed to access the SLB instance	SLB listener port	Accept	Allows inbound traffic from specified client IP addresses.
2	Health check protocol	100.64.0.0/10	Health check port	Accept	Allows inbound traffic from health check IP addresses.

- Outbound rules

Effective order	Protocol	Destination IP addresses	Destination port range	Action	Description
1	All	Client IP addresses allowed to access the SLB instance	-1/-1	Accept	Allows all outbound traffic to specified client IP addresses.
2	All	100.64.0.0/10	-1/-1	Accept	Allows outbound traffic to health check IP addresses.

Ephemeral ports

Clients use different ports to initiate requests. You can select different port ranges for network ACL rules based on the client type. The following table lists ephemeral port ranges for common clients.

Client	Port range
Linux	32768/61000
Windows Server 2003	1025/5000
Windows Server 2008 and later	49152/65535
NAT gateway	1024/65535

21.1.7.3. Create a network ACL

A network access control list (ACL) allows you to manage network access in a virtual private cloud (VPC). This topic describes how to create a network ACL in a VPC.

Prerequisites

A VPC is created. For more information, see Create a VPC.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **Network ACL**.
3. In the top navigation bar, select the region where you want to create the network ACL.
4. On the **Network ACL** page, click **Create Network ACL**.
5. In the **Create Network ACL** dialog box, set the following parameters, and click **OK**.

Parameter	Description
Organization	Select the organization to which the network ACL belongs.
Resource Set	Select the resource set to which the network ACL belongs.
Region	Select the region where you want to deploy the network ACL.
Name	Enter a name for the network ACL. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), and hyphens (-). It must start with a letter.
Description	Enter a description for the network ACL. The description must be 2 to 256 characters in length, and can contain letters, digits, underscores (_), colons (:), and hyphens (-). It must start with a letter and cannot start with `http://` or `https://`.
VPC	Select the VPC for which you want to create the network ACL. **Note** The VPC and network ACL must be deployed in the same region.

What's next

- Associate a network ACL with a VSwitch
- Add an inbound rule
- Add an outbound rule

21.1.7.4. Associate a network ACL with a vSwitch

After you create a network access control list (ACL), you can associate the network ACL with a vSwitch. This way, you can use the network ACL to manage the traffic of the Elastic Compute Service (ECS) instances in the vSwitch.

Prerequisites

Before you associate a network ACL with a vSwitch, make sure that the following requirements are met:

- A network ACL is created. For more information, see Work with network ACLs.
- A vSwitch is created. The vSwitch and network ACL belong to the same virtual private cloud (VPC). For more information, see Work with vSwitches.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **Network ACL**.
3. In the top navigation bar, select the region where the network ACL is created.

4. On the **Network ACL** page, find the network ACL that you want to manage and click **Manage** in the **Actions** column.
5. On the **Resources** tab, click **Bind Resource**.
6. In the **Associate vSwitch** panel, select the vSwitch and click **OK**.

> **Note** The network ACL and vSwitch must belong to the same VPC. A vSwitch can be associated with only one network ACL.

What's next

- Add an inbound rule
- Add an outbound rule

21.1.7.5. Add network ACL rules

21.1.7.5.1. Add an inbound rule

This topic describes how to add an inbound rule to a network access control list (ACL). You can use inbound rules to manage whether Elastic Compute Service (ECS) instances in a vSwitch can be accessed over the Internet or private networks.

Prerequisites

A network ACL is created. For more information, see Work with network ACLs.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **Network ACL**.
3. In the top navigation bar, select the region where the network ACL is created.
4. On the **Network ACL** page, find the network ACL and click **Inbound Rule** in the **Actions** column.
5. On the **Inbound Rule** tab, click **Create Inbound Rule**.
6. In the **Create Inbound Rule** panel, set the following parameters and click **OK**.

Parameter	Description
Name	Enter a name for the inbound rule. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), and hyphens (-). The name must start with a letter and cannot start with `http://` or `https://`.
Effective order	The order in which the inbound rule takes effect. Valid values: 1 to 20. A smaller number indicates a higher priority. For more information, see Rule descriptions.
Action	Select an action for the inbound rule. Valid values: ○ **Accept**: ECS instances in the vSwitch can be accessed. ○ **Drop**: ECS instances in the vSwitch cannot be accessed.

Parameter	Description
Protocol	Select a Layer 4 protocol. Valid values: ○ **ALL**: all protocols. ○ **ICMP**: Internet Control Message Protocol (ICMP). ○ **GRE**: Generic Routing Encapsulation (GRE). ○ **TCP**: Transmission Control Protocol (TCP). ○ **UDP**: User Datagram Protocol (UDP).
Source IP Address	The destination CIDR block to which data is transmitted. Default value: 0.0.0.0/32.
Destination Port Range	Enter the destination port range. Valid values: 1 to 65535. Separate the first port and last port with a forward slash (/), such as 1/200 or 80/80. A value of -1/-1 indicates that all ports are available. Therefore, do not set the value to -1/-1.

21.1.7.5.2. Add an outbound rule

This topic describes how to add an outbound rule to a network access control list (ACL). You can use outbound rules to manage whether Elastic Compute Service (ECS) instances in a vSwitch can access the Internet or private networks.

Prerequisites

A network ACL is created. For more information, see Work with network ACLs.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **Network ACL**.
3. In the top navigation bar, select the region where the network ACL is created.
4. On the **Network ACL** page, find the network ACL and click **Outbound Rule** in the **Actions** column.
5. On the **Outbound Rule** tab, click **Create Outbound Rule**.
6. In the **Create Outbound Rule** pane, set the following parameters and click **OK**.

Parameter	Description
Name	Enter a name for the outbound rule. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), and hyphens (-). The name must start with a letter and cannot start with `http://` or `https://`.
Effective order	The order in which the outbound rule takes effect. Valid values: 1 to 20. A smaller number indicates a higher priority. For more information, see Rule descriptions.

Parameter	Description
Action	Select an action for the outbound rule. Valid values: ○ **Accept**: ECS instances in the vSwitch are allowed to access the Internet or private networks. ○ **Drop**: ECS instances in the vSwitch are not allowed to access the Internet or private networks.
Protocol	Select a Layer 4 protocol. Valid values: ○ **ALL**: all protocols. ○ **ICMP**: Internet Control Message Protocol (ICMP). ○ **GRE**: Generic Routing Encapsulation (GRE). ○ **TCP**: Transmission Control Protocol (TCP). ○ **UDP**: User Datagram Protocol (UDP).
Destination IP Addresses	The destination CIDR block to which data is transmitted. Default value: 0.0.0.0/32.
Destination Port Range	Enter the destination port range. Valid values: 1 to 65535. Separate the first port and last port with a forward slash (/), such as 1/200 or 80/80. -1/-1 indicates that all ports are available. Therefore, do not set the value to -1/-1.

21.1.7.5.3. Modify the priority of a network ACL rule

Network access control list (ACL) rules take effect in descending order of priority. A smaller number indicates a higher priority. This topic describes how to modify the priority of a network ACL rule.

Modify the priority of an inbound rule

1. Log on to the VPC console.
2. In the left-side navigation pane, click **Network ACL**.
3. In the top navigation bar, select the region where the network ACL is created.
4. On the **Network ACL** page, find the network ACL and click **Manage** in the **Actions** column.
5. Click the **Inbound Rule** tab and click **Sort**.
6. In the **Sort** panel, drag rules up or down and click **OK**.

> **Note** An upper rule has a higher priority.

Adjust the priority of an outbound rule

1. Log on to the VPC console.
2. In the left-side navigation pane, click **Network ACL**.
3. In the top navigation bar, select the region where the network ACL is created.
4. On the **Network ACL** page, find the network ACL and click **Manage** in the **Actions** column.
5. Click the **Outbound Rule** tab and click **Sort**.
6. In the **Sort** panel, drag rules up or down and click **OK**.

> **Note** An upper rule has a higher priority.

Related information

- Add an inbound rule
- Add an outbound rule

21.1.7.6. Disassociate a network ACL from a vSwitch

This topic describes how to disassociate a network access control list (ACL) from a vSwitch. After you disassociate a network ACL from a vSwitch, the network ACL no longer controls the traffic of Elastic Compute Service (ECS) instances that belong to the vSwitch.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **Network ACL**.
3. In the top navigation bar, select the region where the network ACL is created.
4. On the **Network ACL** page, find the network ACL and click **Manage** in the **Actions** column.
5. On the **Resources** tab, find the vSwitch and click **Unbind** in the **Actions column**.
6. In the **Unbind Network ACL** message, click **OK**.

21.1.7.7. Delete a network ACL

This topic describes how to delete a network access control list (ACL).

Prerequisites

Make sure that the network ACL is not associated with a vSwitch. If the network ACL is associated with a vSwitch, disassociate the network ACL from the vSwitch first. For more information, see Disassociate a VSwitch from a network ACL.

Procedure

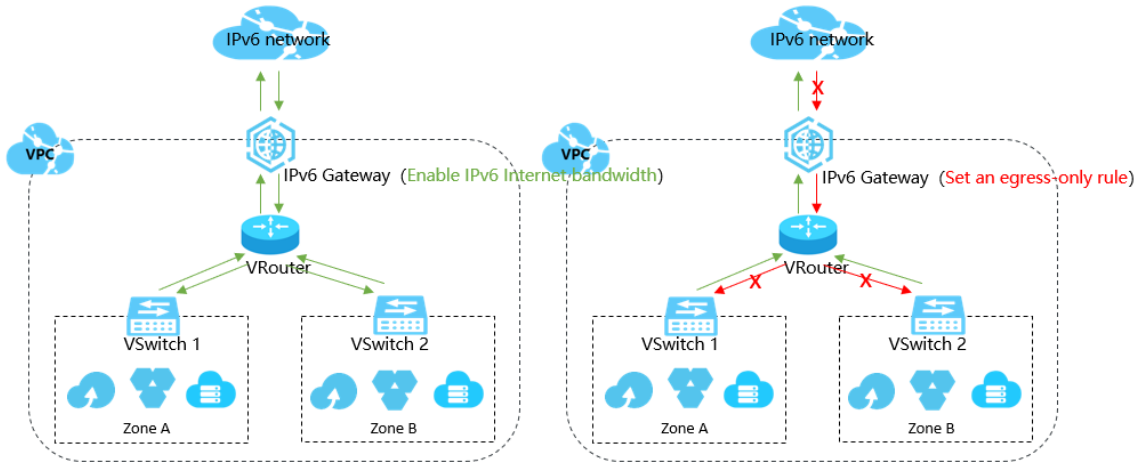
1. Log on to the VPC console.
2. In the left-side navigation pane, click **Network ACL**.
3. In the top navigation bar, select the region where the network ACL is created.
4. On the **Network ACL** page, find the network ACL and click **Delete** in the **Actions** column.
5. In the **Delete Network ACL** message, click **OK**.

22.IPv6 Gateway

22.1. User Guide

22.1.1. What is an IPv6 Gateway?

This topic provides an overview of the IPv6 Gateways of Virtual Private Cloud (VPC). An IPv6 Gateway functions as an IPv6 traffic gateway for a VPC. You can configure the IPv6 Internet bandwidth and egress-only rules to manage the inbound and outbound IPv6 traffic.



Functions

The functions of an IPv6 gateway are as follows:

- **IPv6 internal network communication**

  By default, an IPv6 address in a VPC is allocated with an Internet bandwidth of 0 Mbit/s and only supports communication over the internal network. Specifically, the cloud instances in a VPC can only access other IPv6 addresses in the same VPC through the IPv6 address. The resources cannot access the Internet with these IPv6 addresses or be accessed by IPv6 clients over the Internet.

- **IPv6 public network communication**

  You can purchase an Internet bandwidth for the IPv6 address for which you have applied. In this way, the resources in the VPC can access the Internet through the IPv6 address and be accessed by IPv6 clients over the Internet.

  You can set the Internet bandwidth to 0 Mbit/s at any time to deny the IPv6 address Internet access. After this configuration, the IPv6 address can only communicate over the internal network.

- **IPv6 public network communication with an egress-only rule**

  You can set an egress-only rule for an IPv6 Gateway. In this way, the IPv6 address can access the Internet, but IPv6 clients are denied access to your cloud resources in the VPC over the Internet.

  You can delete the egress-only rule at any time. After the rule is deleted, your resources in the VPC can access the Internet through the IPv6 address for which you have purchased Internet bandwidth, and IPv6 clients can access the resources in the VPC over the Internet.

The network access capability of IPv6 addresses is dependent on the settings of the network type, Internet bandwidth, and egress-only rule, as shown in the following table.

IPv6 network type	Enable IPv6 Internet bandwidth?	Set an egress-only rule?	IPv6 network access capability
Internal network	No	No	Internal network communication
Public network	Yes	No	Internal network communication Public network communication
		Yes	Internal network communication Public network communication when access is initiated by VPCs

Benefits

IPv6 Gateway provides the following benefits:

- **High availability**

  IPv6 Gateways provide cross-zone high availability and stable IPv6 Internet gateway services.
- **High performance**

  A single IPv6 Gateway provides a 10-gigabit level throughput.
- **Flexible management of public network communication**

  You can manage the Internet communication capability of an IPv6 Gateway by adjusting its Internet bandwidth and setting an egress-only rule.

22.1.2. Log on to the IPv6 Gateway console

This topic describes how to log on to the Apsara Uni-manager Management Console to manage your IPv6 gateways. The Google Chrome browser is used as an example.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

> **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
>
> - Uppercase or lowercase letters
> - Digits
> - Special characters, which include ! @ # $ %

3. Click **Login**.
4. In the top navigation bar, choose **Products > Networking > Virtual Private Cloud**.
5. In the left-side navigation pane, choose **Internet Access > IPv6 Gateway**.

22.1.3. Quick start

22.1.3.1. Create an IPv6 VPC

This topic describes how to create a virtual private cloud (VPC) that supports IPv6 CIDR blocks and then create an Elastic Compute Service (ECS) instance that is assigned an IPv6 address in the VPC to access IPv6 services.

Step 1: Create a VPC and a vSwitch

Before you deploy cloud resources in a VPC, you must create a VPC and a vSwitch.

Perform the following steps to create a VPC and a vSwitch:

1. Log on to the VPC console.
2. On the **VPCs** page, click **Create VPC**.
3. On the **Create VPC** page, set the following parameters to configure the VPC and click **OK**.

Parameter	Description
Organization	Select the organization to which the VPC belongs.
Resource Set	Select the resource set to which the VPC belongs.
Region	Select the region where you want to deploy the VPC.
Sharing Scope	Specify the scope of entities that are allowed to use the VPC. ○ **Current Resource Set**: If you select this option, the administrator of the current resource set can create resources in the VPC. ○ **Current Organization and Subordinate Organizations**: If you select this option, administrators that belong to the current organization and subordinate organizations can create resources in the VPC. ○ **Current Organization**: If you select this option, administrators that belong to the current organization can create resources in the VPC. In this example, **Current Resource Set** is selected.

Parameter	Description
VPC Name	Enter a name for the VPC. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character and cannot start with `http://` or `https://`. In this example, **VPCtest** is entered.
IPv4 CIDR Block	Specify the IPv4 CIDR block of the VPC. You can specify an IPv4 CIDR block in one of the following ways: ○ **Recommended CIDR Block**: You can use one of the following standard IPv4 CIDR blocks: 192.168.0.0/16, 172.16.0.0/12, and 10.0.0.0/8. ○ **Custom CIDR Block**: Enter 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, or a subset of these CIDR blocks. The subnet mask must be 8 to 28 bits in length. For example, you can enter 192.168.0.0/16. In this example, Recommended CIDR Block is selected and 192.168.0.0/16 is selected as the IPv4 CIDR block of the VPC. **Note** After you create a VPC, you cannot change its IPv4 CIDR block.
IPv6 CIDR Block	Specify whether to assign an IPv6 CIDR block. ○ **Do Not Assign**: If you select this option, the system does not assign an IPv6 CIDR block to the VPC. ○ **Assign**: If you select this option, the system automatically assigns an IPv6 CIDR block to the VPC. In this example, **Assign** is selected.
Description	Enter a description for the VPC. The description must be 2 to 256 characters in length and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character. It cannot start with `http://` or `https://`.

4. Click **Back to Console**. In the left-side navigation pane, click **VSwitches**.
5. On the **VSwitches** page, click **Create VSwitch**.
6. On the **vSwitch** page, set the following parameter to configure the vSwitch and click **Submit**.

Parameter	Description
Organization	Select the organization to which the vSwitch belongs.
Resource Set	Select the resource set to which the vSwitch belongs.
Region	Select the region where you want to deploy the vSwitch.
Zone	Select the zone where you want to deploy the vSwitch. In a VPC, each vSwitch can be deployed in only one zone. You cannot deploy a vSwitch across zones. However, you can deploy cloud resources in vSwitches that belong to different zones to achieve zone-disaster recovery. **Note** Each cloud resource can be added to only one vSwitch.

Parameter	Description
Sharing Scope	Specify the scope of entities that are allowed to use the vSwitch. ○ **Current Resource Set**: If you select this option, the administrator of the current resource set can create resources in the vSwitch. ○ **Current Organization and Subordinate Organizations**: If you select this option, administrators that belong to the current organization and subordinate organizations can create resources in the vSwitch. ○ **Current Organization**: If you select this option, administrators that belong to the current organization can can create resources in the vSwitch. In this example, **Current Resource Set** is selected.
VPC	Select the VPC where you want to deploy the vSwitch. In this example, VPCtest is selected.
Dedicated for Out-of-cloud Physical Machines	Specify whether the vSwitch to be created is dedicated to bare-metal servers. For more information about bare-metal servers, see the **Bare-metal servers in VPCs** topic in *Bare-metal Server Management Service User Guide*. In this example, **No** is selected.
vSwitch Name	Enter a name for the vSwitch. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character. It cannot start with `http://` or `https://`.
IPv4 CIDR Block	Enter an IPv4 CIDR block for the vSwitch. In this example, the default IPv4 CIDR block is used.
IPv6 CIDR Block	Enter an IPv6 CIDR block for the vSwitch. In this example, the default IPv6 CIDR block is used.
Description	Enter a description for the vSwitch. The description must be 2 to 256 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character. It cannot start with `http://` or `https://`.

Step 2: Create a security group

Perform the following steps to create a security group:

1. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.
2. Choose **Networks and Security > Security Groups**.
3. On the **Security Groups** page, click **Create Security Group**.
4. On the **Create Security Group** page, set the following parameters to configure the security group and click **Submit**.

Parameter	Description
Organization	Select the organization to which the security group belongs.

Parameter	Description
Resource Set	Select the resource set to which the security group belongs.
Region	Select the region to which the security group belongs. Make sure that the security group and the VPC belong to the same region.
Zone	Select the zone to which the security group belongs.
VPC	Select the VPC to which the security group belongs.
Security Group Name	Enter a name for the security group. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character. It cannot start with `http://` or `https://`.
Description	Enter a description for the security group. The description must be 2 to 256 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character. It cannot start with `http://` or `https://`.

Step 3: Create and configure an ECS instance

After you create a VPC and a vSwitch, you must create an ECS instance and assign an IPv6 address to the ECS instance. You must associate this IPv6 address with the network interface controller (NIC) of the ECS instance.

Perform the following steps to create and configure an ECS instance:

1. In the top navigation bar, choose **Products > Networking > Virtual Private Cloud**.
2. In the left-side navigation pane, click **VSwitches**.
3. Select the region where the vSwitch is created.
4. On the **VSwitches** page, find the vSwitch that you want to manage and choose **Create > ECS Instance** in the **Actions** column.
5. On the **Create ECS Instance** page, configure the ECS instance and click **Submit**.

   In this example, **Assign** is selected. Therefore, an IPv6 IP address is assigned to the ECS instance. For more information about other parameters that you are required to specify when you create an ECS instance, see **Create an ECS instance** in **Quick Start** of *Elastic Compute Service User Guide*.
6. Return to the **Instances** page and click the instance ID to view the IPv6 address that is assigned to the ECS instance.
7. Configure a static IPv6 address.
   - If the image of your ECS instance supports DHCPv6, you do not need to manually configure a static IPv6 address. DHCPv6 enables automatic configuration of IPv6 addresses. Therefore, if your ECS instance image supports DHCPv6, the ECS instance can use the assigned IPv6 address to communicate within the private network.

     The following images support DHCPv6:

     - Linux images:
       - CentOS 7.6 IPV6 64Bit
       - CentOS 6.10 64Bit
       - SUSE Linux Enterprise Server 12 SP4 64Bit

- Windows Server images

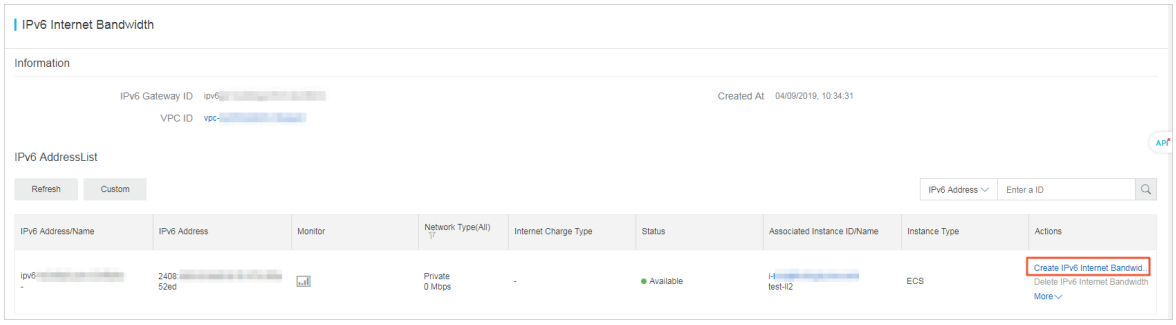
- If the image of your ECS instance does not support DHCPv6, you must manually configure an IPv6 address for the ECS instance. We recommend that you refer to the related documentation for each image for configuration guidance.

Step 4: Purchase an IPv6 Internet bandwidth plan

By default, IPv6 addresses are only used for communication within private networks. If you want to allow an instance that is assigned an IPv6 address to access the Internet or receive requests from IPv6 clients over the Internet, you must purchase an Internet bandwidth plan for the IPv6 address.

Perform the following steps to purchase an Internet bandwidth plan for the IPv6 address:

1. In the top navigation bar, choose **Products > Networking > IPv6 Gateway**.
2. Select the region where the IPv6 gateway is created.
3. On the **IPv6 Gateway** page, find the IPv6 gateway that you want to manage and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **IPv6 Internet Bandwidth**.
5. On the **IPv6 Internet Bandwidth** page, find the IPv6 address that you want to manage and click **Enable IPv6 Internet Bandwidth** in the **Actions** column.



6. Select a bandwidth plan and click **OK**.

   The maximum IPv6 Internet bandwidth for an IPv6 gateway of the Free, Enterprise, or Enhanced Edition is 2 Gbit/s.

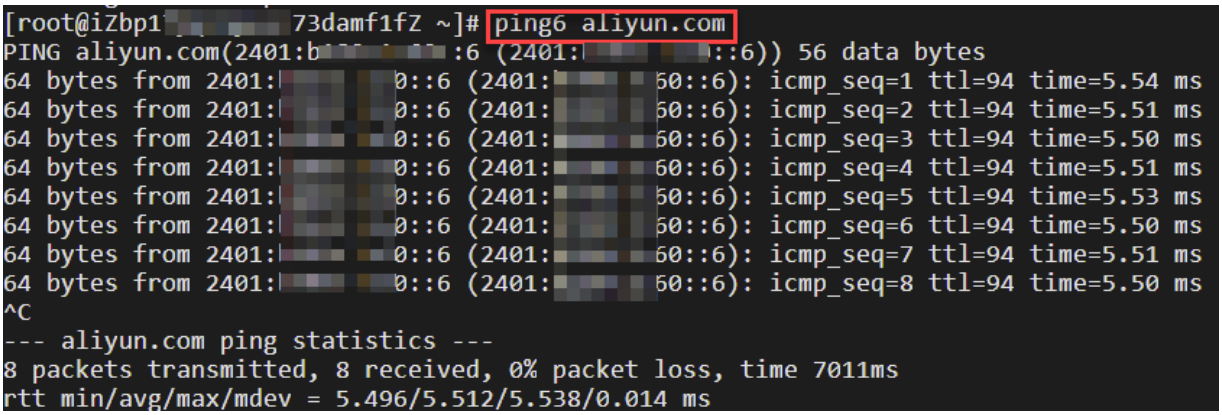
Step 5: Configure security group rules

IPv4 and IPv6 addresses are independent of each other. If the current security group rules do not apply to your IPv6 services, you must configure security group rules for the ECS instances to regulate communication with IPv6 addresses.

For more information about how to configure security rules, see the **Add security group rules** chapter in **Security Groups** of *Elastic Compute Service User Guide*.

Step 6: Test the network connectivity

Log on to an ECS instance and **ping** an IPv6 service to test the network connectivity.



22.1.4. Enable IPv6 for VPCs

22.1.4.1. Create an IPv4 and IPv6 dual-stack VPC

This topic describes how to configure both IPv4 and IPv6 CIDR blocks for a VPC when you create the VPC. By default, all VPCs are associated with IPv4 CIDR blocks which cannot be deleted. However, you can choose whether to allocate an IPv6 CIDR blocks to a VPC. After you choose to allocate an IPv6 CIDR block to a VPC, the system creates an IPv6 gateway of the Free Edition for the VPC for you to provision IPv6 bandwidth and manage IPv6 traffic.

Procedure

1. Log on to the VPC console.
2. On the top of the page, select a region to deploy your VPC.
3. On the **VPCs** page, click **Create VPC**.
4. On the **Create VPC** page, configure the VPC network and click **Submit**. The following table describes the configuration parameters.

Parameter	Description
Organization	Select the organization to which the VPC belongs.
Resource Set	Select the resource set to which the VPC belongs.
Region	Select a region to deploy the VPC.
Share with Sub-organizations	Specify whether to share the VPC. If you select Yes, administrators of sub-organizations can create resources in the VPC network.
VPC Name	Enter a name for the VPC. The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.

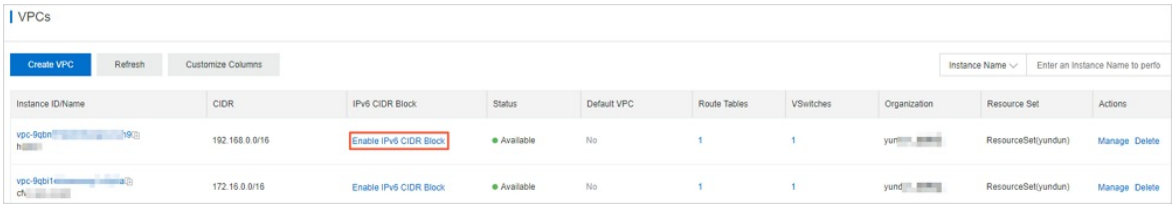
Parameter	Description
IPv4 CIDR Block	Specify an IPv4 CIDR block for the VPC. The following setting methods are supported: ○ **Recommended CIDR Block**: Use 192.168.0.0/16, 172.16.0.0/12, or 10.0.0.0/8 as the IPv4 CIDR block of the VPC. ○ **Custom CIDR Block**: Use 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, or one of their subnets as the IPv4 CIDR block of the VPC. The subnet mask must be 8 to 24 bits in length. For example, enter 192.168.0.0/16. **Note** After you create a VPC, you cannot modify its IPv4 CIDR block.
IPv6 CIDR Block	Specify whether to assign an IPv6 CIDR block to the VPC. In this example, select **Assign**. If you set this parameter to Assign, the system automatically creates an IPv6 gateway of the Free Edition for your VPC and assigns an IPv6 CIDR block with the subnet mask /61, such as 2xx1:db8::/61. By default, IPv6 addresses can only be used for communication within private networks. If you need to enable an IPv6 address to access and be accessed over the Internet, you must purchase IPv6 Internet bandwidth for the IPv6 address. For more information, see Enable Internet connectivity for an IPv6 address. **Note** After you create a VPC, you cannot modify its IPv6 CIDR block.
Description	Enter a description for the VPC. The description must be 2 to 256 characters in length and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.

22.1.4.2. Enable an IPv6 CIDR block for a VPC network

This topic describes how to enable an IPv6 CIDR block for a Virtual Private Cloud (VPC) network. After the IPv6 CIDR block is enabled, the system automatically creates an IPv6 gateway free of charge for the VPC network. You can use the IPv6 Gateway to manage the IPv6 Internet bandwidth and set egress-only rules.

Procedure

1. Log on to the VPC console.
2. Select the region where your VPC network is deployed.
3. On the **VPCs** page, find the target VPC network and click **Enable IPv6 CIDR Block** in the **IPv6 CIDR Block** column.



4. In the **Enable IPv6 CIDR Block** dialog box, select **Enable IPv6 CIDR Block of all VSwitches in VPC**, and then click **OK**.

   If you do not select **Enable IPv6 CIDR Block of all VSwitches in VPC**, you must enable IPv6 CIDR Block for each VSwitch. For more information, see Enable IPv6 for a vSwitch.

22.1.5. Enable IPv6 for vSwitches

22.1.5.1. Create an IPv4 and IPv6 dual-stack VSwitch

This topic describes how to assign an IPv6 CIDR block to a VSwitch when you create the VSwitch.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **VSwitches**.
3. Select the region where you want to deploy the VSwitch.
4. On the **VSwitches** page, click **Create VSwitch**.
5. On the **VSwitch** page, configure the VSwitch and click **Submit**. The following table describes the configuration parameters.

Parameter	Description
Organization	Select the organization to which the VSwitch belongs.
Resource Set	Select the resource set to which the VSwitch belongs.
Region	Select a region to deploy the VSwitch.
Zone	Select a zone to deploy the VSwitch. Each VSwitch must reside entirely within one zone and cannot span multiple zones. However, you can deploy cloud resources in VSwitches of different zones to achieve cross-zone disaster recovery. **Note** Each cloud resource can be deployed in only one VSwitch.
VPC	Select the VPC for which you want to create the VSwitch.
Dedicated for Off-Cloud Servers	Specify whether the VSwitch is dedicated for off-cloud servers. For more information, see *the* **Features of off-cloud servers for VPC** topic in the Apsara Stack Bare Metal Server User Guide.
VSwitch Name	Enter a name for the VSwitch. The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.

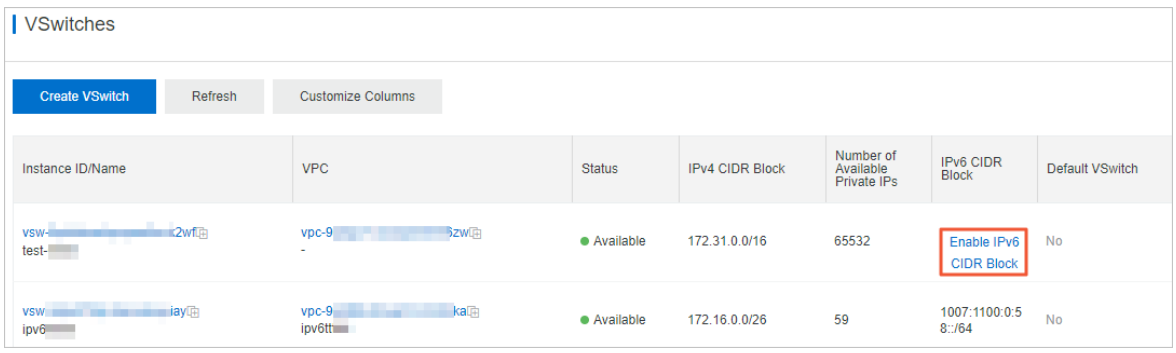
Parameter	Description
IPv4 CIDR Block	Specify an IPv4 CIDR block for the VSwitch. ○ You must specify the IP address range for the VSwitch in the form of a CIDR block. The IPv4 CIDR block size for a VSwitch is between a 16-bit mask and a 29-bit mask. It means that 8 to 65,536 IP addresses can be provided. ○ The IPv4 CIDR block of a VSwitch must be a subset of the IPv4 CIDR block of the VPC this VSwitch resides in. ○ The first and the last three IP addresses of each VSwitch IPv4 CIDR block are reserved. For example, if the VSwitch IPv4 CIDR block is 192.168.1.0/24, the IP addresses 192.168.1.0, 192.168.1.253, 192.168.1.254, and 192.168.1.255 are reserved. ○ The IPv4 CIDR block of a VSwitch must be more specific than the CIDR range of a route in any of the VPC route tables. ○ After you create a VSwitch, you cannot modify its IPv4 CIDR block.
IPv6 CIDR Block	Specify an IPv6 CIDR block for the VSwitch. The default subnet mask for the IPv6 CIDR block of a VSwitch is /64. You can enter a decimal number ranging from 0 to 255 to define the last 8 bits of the IPv6 CIDR block. For example, if the IPv6 CIDR block of the VPC that contains the VSwitch is 2xx1:db8::/64, you can enter 255 (FF in hexadecimal notation) in this field to define the IPv6 CIDR block of the VSwitch as 2xx1:db8:ff::/64.
Description	Enter a description for the VSwitch. The description must be 2 to 256 characters in length and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.

22.1.5.2. Enable IPv6 for a vSwitch

This topic describes how to allocate an IPv6 CIDR block to a vSwitch.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **vSwitches**.
3. Select the region where your vSwitch resides.
4. On the **vSwitches** page, find the target vSwitch, and click **Enable IPv6 CIDR Block** in the **IPv6 CIDR Block** column.



5. (Optional)In the **Enable IPv6 CIDR Block** dialog box, click **OK**.

> **Note** This operation is required only when IPv6 is not enabled for the VPC to which the vSwitch belongs.

6. Specify an IPv6 CIDR block and click **OK**.

   The default subnet mask for the IPv6 CIDR block of a vSwitch is /64. You can enter a decimal number ranging from 0 to 255 to define the last 8 bits of the IPv6 CIDR block.

   For example, if the IPv6 CIDR block of the VPC that contains the vSwitch is 2xx1:db8::/64, you can enter 255 (FF in hexadecimal notation) in this field to define the IPv6 CIDR block of the vSwitch as 2xx1:db8::ff/64.

22.1.6. Manage IPv6 Gateways

22.1.6.1. Editions of IPv6 gateways

This topic describes the different editions of IPv6 gateways. Different quotas and limits apply to IPv6 gateways of different editions, such as the maximum forwarding bandwidth, maximum IPv6 bandwidth per IPv6 address, and maximum number of egress-only rules.

IPv6 gateway edition	Maximum forwarding bandwidth	Maximum IPv6 bandwidth per IPv6 address	Maximum number of egress-only rules
Free Edition	10 Gbit/s	2 Gbit/s	0
Enterprise Edition	20 Gbit/s	2 Gbit/s	50
Enhanced Enterprise Edition	50 Gbit/s	2 Gbit/s	200

22.1.6.2. Create an IPv6 gateway

This topic describes how to create an IPv6 gateway for a VPC. After you create an IPv6 gateway, you can purchase IPv6 Internet bandwidth and set egress-only rules for the IPv6 gateway.

Prerequisites

Make sure that IPv6 is enabled for the VPC before you create an IPv6 gateway for the VPC. For more information, see Allocate an IPv6 CIDR block when you create a VPC and Enable an IPv6 CIDR block for a VPC network.

Procedure

1. Log on to the IPv6 Gateway console.
2. On the **IPv6 Gateway** page, click **Create IPv6 Gateway**.
3. Configure the IPv6 gateway and click **Submit**. The following table describes the configuration parameters.

Parameter	Description
Organization	Select the organization to which the IPv6 gateway belongs.
Resource Set	Select the resource set to which the IPv6 gateway belongs.
Region	Select a region to deploy the IPv6 gateway. The IPv6 gateway must belong to the same region as the VPC for which you want to create the IPv6 gateway.

Parameter	Description
VPC	Select the VPC for which you want to create an IPv6 gateway. The target VPC may not be available due to the following reasons: ○ Only one IPv6 gateway can be created for each VPC. The VPC already has an IPv6 gateway. ○ The VPC has a custom route with the destination CIDR block set to ::/0. If this happens, you must delete this custom route before you can create an IPv6 gateway for the VPC. **Note** After an IPv6 gateway is created, you cannot change the VPC that is associated with the IPv6 gateway.
Edition	Select the edition of the IPv6 gateway. IPv6 gateways are available in the following editions: ○ Free Edition ○ Enterprise Edition ○ Enhanced Enterprise Edition Different quotas and limits apply to IPv6 gateways of different editions, such as the maximum forwarding bandwidth, maximum IPv6 bandwidth per IPv6 address, and maximum number of egress-only rules. For more information, see Editions of IPv6 gateways.

22.1.6.3. Modify an IPv6 gateway

This topic describes how to modify the name and description of an IPv6 gateway.

Procedure

1. Log on to the IPv6 Gateway console.
2. Select the region where the IPv6 gateway resides.
3. On the **IPv6 Gateway** page, find the target IPv6 gateway, and click **Manage** in the **Actions** column.
4. On the **IPv6 Gateway Details** page, click **Edit** next to **Name**. In the dialog box that appears, enter a new name for the IPv6 gateway, and click **OK**.

   The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), and hyphens (-). It must start with a letter.
5. Click **Edit** next to **Description**. In the dialog box that appears, enter a new description for the IPv6 gateway and click **OK**.

   The description must be 2 to 256 characters in length and cannot start with `http://` or `https://`.

22.1.6.4. Delete an IPv6 gateway

This topic describes how to delete an IPv6 gateway. If a VPC no longer needs to access or be accessed by IPv6 clients, you can delete the IPv6 gateway associated with the VPC.

Prerequisites

Before you delete an IPv6 gateway of the enterprise edition or enhanced enterprise edition, you must delete the egress-only rules of the IPv6 gateway. For more information, see Delete an egress-only rule.

Procedure

1. Log on to the IPv6 Gateway console.
2. Select the region where the IPv6 gateway resides.
3. On the **IPv6 Gateway** page, find the target IPv6 gateway, and click **Delete** in the **Actions** column.
4. In the dialog box that appears, click **OK**.

22.1.7. Manage IPv6 Internet bandwidth

22.1.7.1. Enable Internet connectivity for an IPv6 address

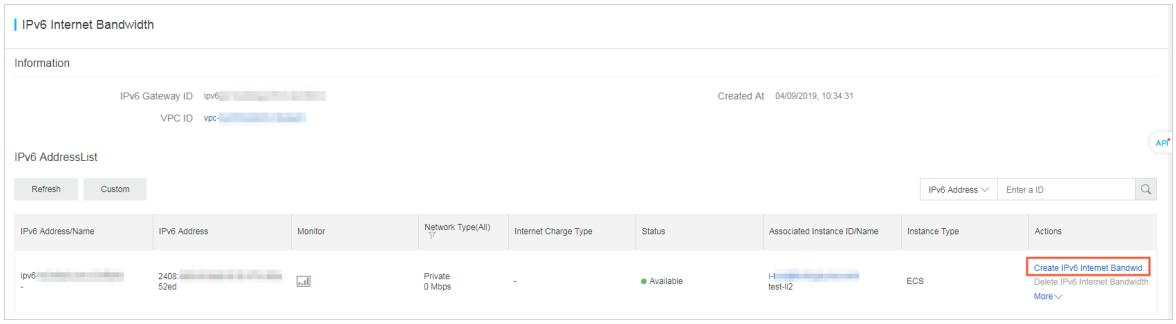
This topic describes how to enable Internet bandwidth for an IPv6 address. After Internet bandwidth is enabled for an IPv6 address, the IPv6 address can be used to communicate over the Internet.

Prerequisites

An ECS instance is created in the VPC associated with the corresponding IPv6 gateway and is configured with an IPv6 address.

Procedure

1. Log on to the IPv6 Gateway console.
2. Select the region where the IPv6 gateway resides.
3. On the **IPv6 Gateway** page, find the target IPv6 gateway, and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **IPv6 Internet Bandwidth**.
5. On the **IPv6 Internet Bandwidth** page, find the target IPv6 address, and click **Enable IPv6 Internet Bandwidth** in the **Actions** column.



6. Specify a bandwidth and click **Submit**.

   An IPv6 gateway of the Free Edition, Enterprise Edition, or Enhanced Enterprise Edition supports a 2 Gbit/s maximum bandwidth per IPv6 address.

22.1.7.2. Modify the maximum bandwidth of an IPv6 address

This topic describes how to modify the maximum bandwidth of an IPv6 address. The modification takes effect immediately.

Procedure

1. Log on to the IPv6 Gateway console.
2. Select the region where the IPv6 gateway resides.
3. On the **IPv6 Gateway** page, find the target IPv6 gateway, and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **IPv6 Internet Bandwidth**.
5. In the **IPv6 AddressList** section, find the target IPv6 address, and choose **More > Modify IPv6 Internet Bandwidth** in the **Actions** column.

6. Specify a bandwidth and click **Submit**.

22.1.7.3. Disable Internet connectivity for an IPv6 address

This topic describes how to delete the Internet bandwidth of an IPv6 address that is no longer needed for Internet communication.

Procedure

1. Log on to the IPv6 Gateway console.
2. Select the region where the IPv6 gateway resides.
3. On the **IPv6 Gateway** page, find the target IPv6 gateway, and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **IPv6 Internet Bandwidth**.
5. In the **IPv6 AddressList** section, find the target IPv6 address, and click **Delete IPv6 Internet Bandwidth** in the **Actions** column.
6. In the dialog box that appears, click **OK**.

22.1.8. Manage egress-only rules

22.1.8.1. Create an egress-only rule

This topic describes how to create an egress-only rule. If you want an instance in the VPC associated with an IPv6 gateway to be able to access the Internet with an IPv6 address, while resources on the Internet cannot initiate communication with this instance, you can create an egress-only rule for the instance.

Prerequisites

IPv6 Internet bandwidth is enabled for the IPv6 address configured for the instance. For more information, see Enable and manage IPv6 Internet bandwidth.

Context

IPv6 gateways of the Free Edition do not support egress-only rules. IPv6 gateways of the enterprise edition and enterprise enhanced edition support a maximum of 50 and 200 egress-only rules respectively.

Procedure

1. Log on to the IPv6 Gateway console.
2. Select the region where the IPv6 gateway resides.
3. On the **IPv6 Gateway** page, find the target IPv6 gateway, and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **Egress-only Rule**.
5. On the **Egress-only Rule** page, click **Create Egress-only Rule**.
6. In the **Create Egress-only Rule** dialog box, select the ECS instance that uses the IPv6 address to communicate with the Internet, and click **OK**.

22.1.8.2. Delete an egress-only rule

This topic describes how to delete an egress-only rule. After the rule is deleted, the IPv6 address with Internet bandwidth can access the Internet, and the instance associated with the IPv6 address can be accessed over the Internet.

Procedure

1. Log on to the IPv6 Gateway console.

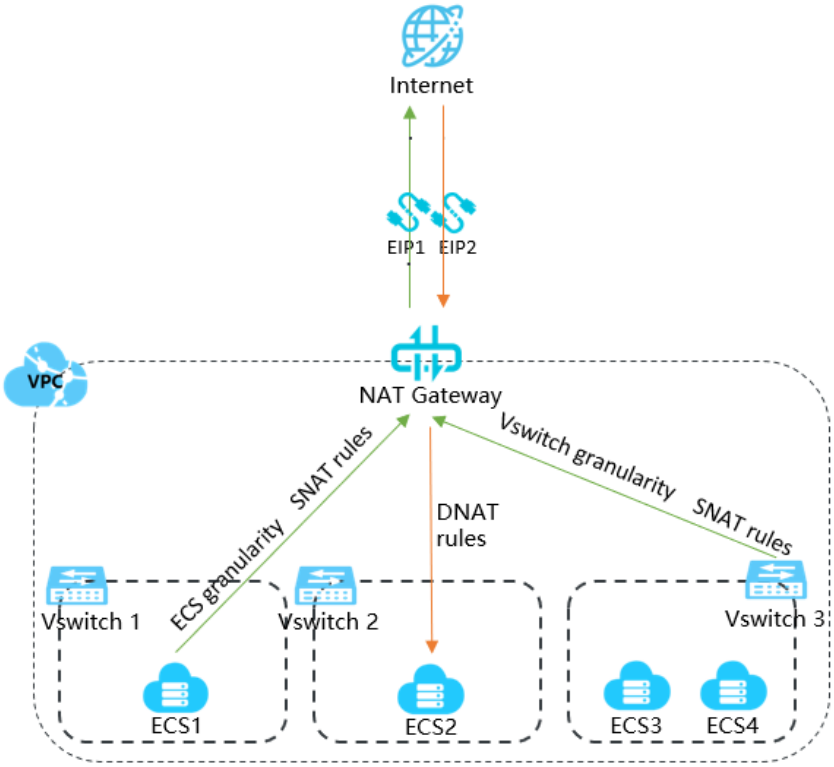
2. Select the region where the IPv6 gateway resides.
3. On the **IPv6 Gateway** page, find the target IPv6 gateway, and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **Egress-only Rule**.
5. On the **Egress-only Rule** page, find the egress-only rule that you want to delete, and click **Delete** in the **Actions** column.
6. In the dialog box that appears, click **OK**.

23.NAT Gateway

23.1. User Guide

23.1.1. What is NAT Gateway?

A NAT gateway is an enterprise-grade Internet gateway. NAT Gateway provides source network address translation (SNAT) and destination network address translation (DNAT) features, a maximum forwarding capacity of 10 Gbit/s, and support for cross-zone disaster recovery.



Features

NAT gateways must be associated with public IP addresses. After you create a NAT gateway, you can associate it with one or more elastic IP addresses (EIPs).

NAT Gateway supports SNAT and DNAT.

- SNAT allows Elastic Compute Service (ECS) instances that are deployed in a virtual private cloud (VPC) and not associated with public IP addresses to access the Internet.
- DNAT maps public IP addresses of a NAT gateway to ECS instances so that the ECS instances can be accessible from the Internet.

23.1.2. Log on to the NAT Gateway console

This topic describes how to log on to the Apsara Uni-manager Management Console to manage your NAT gateways. The Google Chrome browser is used as an example.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Login**.
4. In the top menu bar, choose **Products > Networking > Virtual Private Cloud**.
5. In the left-side navigation pane, choose **Internet Access > NAT Gateway**.

23.1.3. Quick Start

23.1.3.1. Overview

This topic describes how to configure Source Network Address Translation (SNAT) and Destination Network Address Translation (DNAT). You can configure SNAT and DNAT to enable ECS instances in a Virtual Private Cloud (VPC) network to communicate with the Internet through a NAT gateway.

Prerequisites

Before you start, make sure that the following conditions are met:

- A VPC network is created. For more information, see *VPC User Guide* **Create a VPC network** in the **Quick Start** chapter in the VPC User Guide.
- An ECS instance is created in the VPC network. For more information, see *ECS User Guide* **Create an instance** in the **Quick Start** chapter in the ECS User Guide.
- An elastic IP address (EIP) is created. For more information, see *EIP User Guide* **Create an EIP** in the **Quick Start** chapter in the EIP User Guide.

Procedure

In this topic, an ECS instance that is not associated with any public IP addresses in a VPC network is used as an example. The following flowchart shows how to associate an EIP with a NAT gateway:

1. Create a NAT gateway

   A NAT gateway is an enterprise-class gateway that provides NAT proxy services. Before you configure SNAT and DNAT entries, you must create a NAT gateway.

   For more information, see Create a NAT gateway.

2. Associate an EIP to a NAT gateway

   A NAT gateway functions as expected only after it is associated with a public IP address. After you create a NAT gateway, you can associate it with an EIP.

   For more information, see Associate an EIP with a NAT Gateway.

3. Create a DNAT entry

   This topic describes how to create a Destination Network Address Translation (DNAT) entry. DNAT maps public IP addresses to Elastic Compute Service (ECS) instances in a Virtual Private Cloud (VPC) network. This way, the ECS instances can receive requests sent over the Internet. DNAT supports port mapping and IP mapping.

   For more information, see Create a DNAT entry.

4. Create an SNAT entry

   This topic describes how to create a Source Network Address Translation (SNAT) entry. SNAT allows Elastic Compute Service (ECS) instances in a Virtual Private Cloud (VPC) network to access the Internet without using public IP addresses.

   For more information, see Create a SNAT entry.

23.1.3.2. Create a NAT gateway

A NAT gateway is an enterprise-class gateway that provides NAT proxy services. Before you configure SNAT and DNAT entries, you must create a NAT gateway.

Prerequisites

A VPC network is created. For more information, see *VPC User Guide*Create a VPC network in the Quick Start chapter in the VPC User Guide.

Procedure

1. Log on to the NAT Gateway console.
2. On the **NAT Gateway** page, click **Create NAT Gateway**.
3. Set the parameters for the NAT gateway based on the following information, and then click **Submit**.

Parameter	Description
Organization	The organization to which the NAT gateway belongs.
Resource set	The resource set to which the NAT gateway belongs.
Region	The region where the NAT gateway is deployed.

Parameter	Description
VPC	The VPC network to which the NAT gateway belongs. If you cannot find the target VPC network in the list, perform the following operations: ○ Check whether the VPC network is already associated with a NAT gateway. Each VPC network can be associated with only one NAT gateway. ○ Check whether the VPC network has a custom route entry with the destination CIDR block set to 0.0.0.0/0. If such a custom route entry exists, delete it. ○ Check whether the RAM user is authorized to access the VPC network. If the RAM user is not authorized, contact your Alibaba Cloud account owner to grant permissions.
Specification	Select the size of the NAT gateway. Valid values: ○ **Small**: supports up to 10,000 SNAT connections. ○ **Medium**: supports up to 50,000 SNAT connections. ○ **Large**: supports up to 200,000 SNAT connections. ○ **Super Large**: supports up to 1,000,000 SNAT connections. **Note** The size of a NAT gateway determines the maximum number of SNAT connections, but it does not affect the maximum number of DNAT connections.
Parameter	Enter a name for the NAT gateway. The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character but cannot start with `http://` or `https://`.

23.1.3.3. Associate an EIP with the a NAT gateway

A NAT gateway functions as expected only after it is associated with a public IP address. After you create a NAT gateway, you can associate it with an elastic IP address (EIP).

Context

You can associate an EIP or a NAT service plan with a NAT gateway. However, you can choose only one of them for the NAT gateway. If you want to associate a NAT service plan with a NAT gateway, you must purchase a NAT service plan first. Then, you can configure the SNAT or DNAT feature for the NAT gateway. For more information, see Create a NAT service plan.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateways** page, find the NAT gateway, and choose ⋮ > **Bind Elastic IP Address** in the **Actions** column.
4. In the **Bind Elastic IP Address** dialog box, set the following parameters, and then click **OK**.

Parameter	Description
Usable EIP list	Select the EIP that is used to access the Internet.
VSwitch	Select the VSwitch for which you want to add SNAT entries. After a VSwitch is selected, the system automatically adds an SNAT entry so that Alibaba cloud services in the VSwitch can access the Internet. You can also skip this step and manually add SNAT entries after you associate an EIP with the NAT gateway. For more information, see Create a SNAT entry.

23.1.3.4. Create a DNAT entry

This topic describes how to create a Destination Network Address Translation (DNAT) entry. DNAT maps public IP addresses to Elastic Compute Service (ECS) instances in a Virtual Private Cloud (VPC) network. This way, the ECS instances can receive requests sent over the Internet. DNAT supports port mapping and IP mapping.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateways** page, find the target NAT gateway, and click **Configure DNAT** in the **Actions** column.
4. On the **DNAT Table** page, click **Create DNAT Entry**.
5. On the **Create DNAT Entry** page, set the following parameters, and then click **OK**.

Parameter	Description
Public IP address	Select an available public IP address. **Note** If a public IP address is already used in a SNAT entry, it cannot be used to create a DNAT entry.
Private IP address	Select the ECS instance that uses the DNAT entry to receive requests from the Internet. You can specify the private IP address of the target ECS instance in the following ways: ○ **Auto Fill**: select an ECS instance from the ECS instance or Elastic Network Interface (ENI) list. ○ **Manually Input**: enter the private IP address of the target ECS instance. **Note** The private IP address that you enter must fall in the range of the CIDR block or belong to an in-use ECS instance.

Parameter	Description
Port Settings	Select a DNAT mapping method: ○ **All**: This method uses IP mapping. All requests destined for the public IP address are forwarded to the target ECS instance. ○ **Specific Port**: This method uses port mapping. The NAT gateway forwards requests from the specified protocol and port to the specified port of the target ECS instance. After you select a specific port, specify **Public Port** (the external port for port mapping), **Private Port** (the internal port for port mapping), and **IP Protocol** (the protocol of the ports).
Entry Name	Enter a name for the DNAT entry. The name must be 2 to 128 characters in length and can contain digits, underscores (_), and hyphens (-). It must start with a letter or Chinese character.

23.1.3.5. Create an SNAT entry

This topic describes how to create a Source Network Address Translation (SNAT) entry. You can create SNAT entries on the NAT gateway of a Virtual Private Cloud (VPC) network. Then, the Elastic Compute Service (ECS) instances without public IP addresses assigned in the VPC network can use the SNAT entries to access the Internet.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateways** page, find the target NAT gateway and click **Configure SNAT** in the **Actions** column.
4. On the **SNAT Table** page, click **Create SNAT Entry**.
5. In the **Create SNAT Entry** dialog box, set the following parameters, and then click **OK**.

Parameter	Description
VSwitch Granularity	
VSwitch	Select the VSwitch for which you want to create the SNAT entry in the associated VPC network. All ECS instances attached to the specified VSwitch can access the Internet by using the SNAT entry. **Note** SNAT entries do not take effect on ECS instances that are assigned public IP addresses. For example, an ECS instance may be assigned static public IP address, associated with an elastic IP address (EIP) or has a Destination Network Address Translation (DNAT) IP mapping configured. These ECS instances use the public IP addresses instead of the SNAT entries to access the Internet.
VSwitch CIDR block	The CIDR block of the selected VSwitch.

Parameter	Description
Public IP address	Select the public IP address that is used to access the Internet. You can select multiple public IP addresses to create an SNAT IP address pool. **Note** A public IP address that is already used in a DNAT entry cannot be used to create an SNAT entry.
Entry name	Enter a name for the SNAT entry. The name must be 2 to 128 characters in length and can contain digits, underscores (_), and hyphens (-). It must start with a letter or a Chinese character.
ECS Granularity	
Available ECS Instances	Select the ECS instance for which you want to create the SNAT entry in the associated VPC network. The selected ECS instance can access the Internet by using the specified public IP address. Make sure that the following conditions are met: ○ The ECS instance is in the running state. ○ The ECS instance is not associated with an EIP or assigned a static public IP address.
ECS CIDR Block	Displays the CIDR block of the ECS instance.
Public IP address	Select the public IP address that is used to access the Internet. You can select multiple public IP addresses to create an SNAT IP address pool. The maximum bandwidth for each public IP address in an SNAT IP address pool is 200 Mbit/s. To fully utilize the EIP bandwidth plan and avoid port conflicts caused by insufficient public IP addresses, note the following limits when you add public IP addresses to an SNAT entry: ○ If the maximum bandwidth of the EIP bandwidth plan is 1024 Mbit/s, add at least five public IP addresses to the SNAT entry. ○ For each additional 200 Mbit/s of the peak bandwidth to the EIP bandwidth plan, at least one public IP address must be added to the SNAT entry. **Note** A public IP address that is used in a DNAT entry cannot be used to create an SNAT entry.
Entry Name	Enter a name for the SNAT entry. The name must be 2 to 128 characters in length and can contain numbers, underscores (_), and hyphens (-). The name must start with a letter or Chinese character.

23.1.4. Manage a NAT gateway

23.1.4.1. Sizes of NAT gateways

This topic describes the available sizes of Network Address Translation (NAT) gateways.The available sizes are Small, Middle, Large, and Super Large. The maximum number of SNAT connections and the number of SNAT connections per second (CPS) supported by a NAT gateway are determined by the size of NAT gateway. However, the size of a NAT gateway does not affect the performance of Destination Network Address Translation (DNAT).

Compare NAT gateway sizes

The following table lists different sizes of NAT gateways.

Size	Maximum number of SNAT connections	Number of SNAT CPS
Small	10,000	1,000
Middle	50,000	5,000
Large	200,000	10,000
Super Large	1,000,000	30,000

Limits

When you select a size for a NAT gateway, note the following limits:

- Cloud Monitor monitors only the maximum number of SNAT connections for NAT gateways. It does not monitor the number of new SNAT connections per second.
- The timeout of SNAT connections in a NAT gateway is 900 seconds.
- To avoid the timeout of SNAT connections caused by network congestion and Internet instability, make sure that your applications support automatic reconnection. This ensures higher availability.
- NAT gateways do not support packet fragmentation.
- For the same destination public IP address and port, the number of EIPs associated with a NAT gateway determines the maximum number of connections. Each EIP associated with a NAT gateway supports up to 55,000 connections. If N EIPs are associated with the NAT gateway, the maximum number of connections that the NAT gateway supports is N × 55,000.

23.1.4.2. Create a NAT gateway

A NAT gateway is an enterprise-class gateway that provides NAT proxy services. Before you configure SNAT and DNAT entries, you must create a NAT gateway.

Prerequisites

A VPC network is created. For more information, see *VPC User Guide*Create a VPC network in the Quick Start chapter in the VPC User Guide.

Procedure

1. Log on to the NAT Gateway console.
2. On the **NAT Gateway** page, click **Create NAT Gateway**.
3. Set the following parameters and click **submit**.

Parameter	Description
Organization	Select the organization to which the NAT gateway belongs.
Resource Set	Select the resource group to which the NAT gateway belongs.

Parameter	Description
Region	Select the region where you want to deploy the NAT gateway.
VPC	Select the VPC where you want to deploy the NAT gateway. If you cannot find the VPC in the list, perform the following operations: ○ Check whether the VPC is already associated with a NAT gateway. Each VPC can be associated with only one NAT gateway. ○ Check whether the VPC has a custom route entry whose destination CIDR block is 0.0.0.0/0. If the custom route entry exists, delete it. ○ Check whether the Resource Access Management (RAM) user is authorized to access the VPC. If the RAM user is unauthorized, contact the owner of the Apsara Stack account that created the RAM user to grant permissions.
Sharing Scope	Select the participants who can use the VPC to create resources. ○ **Current Resource Set**: Only the administrator of the current resource set can create resources for the shared VPC. ○ **Current Organization and Subordinate Organization**: Only the administrators of the current organization and its subordinate organization can create resources for the shared VPC. ○ **Current Organization**: Only the administrator of the current organization can create resources for the shared VPC.
Specification	Select the size of the NAT gateway. Valid values: ○ **Small**: supports up to 10,000 SNAT connections. ○ **Medium**: supports up to 50,000 SNAT connections. ○ **Large**: supports up to 200,000 SNAT connections. ○ **Super Large-1**: supports up to 1,000,000 SNAT connections. **Note** The maximum number of SNAT connections is limited by the size of a NAT gateway. However, the gateway size does not affect the maximum number of DNAT connections.
Name	Enter a name for the NAT gateway. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with `http://` or `https://`.

23.1.4.3. Modify a NAT gateway

This topic describes how to modify the name and description of a NAT gateway.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateway** page, find the NAT gateway that you want to manage, and then click **Manage** in the **Actions** column.
4. On the **NAT Gateway Details** tab, click **Edit** next to the name. In the dialog box that appears, enter a new name for the NAT gateway, and then click **OK**.

The name must be 2 to128 characters in length, and can contain digits, underscores (_), and hyphens (-). It must start with a letter or Chinese character.

5. Click **Edit** next to the description. In the dialog box that appears, enter a new description, and then click **OK**.

   The description must be 2 to 256 characters in length. It cannot start with `http://` or `https://`.

23.1.4.4. Delete a NAT gateway

You can delete NAT gateways that are billed on a pay-as-you-go basis. You cannot delete subscription NAT gateways.

Prerequisites

Before you delete a NAT gateway, make sure that the following conditions are met:

- The NAT gateway is not associated with an EIP. If the NAT gateway is associated with an EIP, disassociate the EIP first. For more information, see Disassociate EIPs from a NAT gateway.
- The DNAT table is empty. If the DNAT table contains DNAT entries, delete these entries first. For more information, see Delete a DNAT entry.
- The SNAT table is empty. If the DNAT table contains SNAT entries, delete these entries first. For more information, see Delete a SNAT entry.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateways** page, find the target NAT gateway, and click ⋮ > **Delete** in the **Actions** column.
4. In the dialog box that appears, click **OK**.

> **Note** If you select **Delete (Delete NAT gateway and resources)**, the DNAT and SNAT entries of the NAT gateway are deleted automatically. The EIP associated with the NAT gateway is also disassociated.

23.1.5. Manage EIPs

23.1.5.1. Associate an EIP with a NAT gateway

This topic describes how to associate an EIP with a NAT gateway. NAT gateways must be associated with EIPs so that they can work as expected. After you create a NAT gateway, you can associate it with an EIP.

Prerequisites

Before you associate an EIP with a NAT gateway, make sure that the following conditions are met:

- A NAT gateway is created. For more information, see Create a NAT Gateway.
- An EIP is purchased. For more information, see the **Create an Elastic IP address** topic of the **Quick start** document in the *EIP user guide*.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateway** page, find the NAT gateway with which you want to associate an EIP, and choose ⋮ > **Bind Elastic IP Address** in the **Actions** column.

4. In the **Associate EIP** dialog box, set the following parameters, and click **OK**.

Parameter	Description
Usable EIP List	Select the EIP that is used to communicate with the Internet.
vSwitch	Select the vSwitch to which you want to add SNAT entries. After you select a vSwitch, the system automatically adds SNAT entries to the vSwitch. Then, cloud services in the vSwitch can access the Internet. You can skip this step and manually add SNAT entries after you associate an EIP with the NAT gateway. For more information, see Create a SNAT entry.

23.1.5.2. Disassociate an EIP from a NAT gateway

If your NAT gateway does not need to communicate with the Internet, you can disassociate the EIP from the NAT gateway.

Prerequisites

The EIP that you want to disassociate is not used in any SNAT or DNAT entries.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateways** page, find the NAT gateway, and choose ⋮ > **Unbind Elastic IP Address** in the **Actions** column.
4. In the **Unbind Elastic IP Address** dialog box, select the EIP that you want to disassociate, and click **OK**.

23.1.6. Manage a DNAT table

23.1.6.1. DNAT table overview

NAT Gateway supports the Destination Network Address Translation (DNAT) feature. You can create DNAT entries to map public IP addresses to ECS instances in a Virtual Private Cloud (VPC) network. This way, the ECS instances can receive requests from the Internet.

DNAT entries

You can configure port mapping when you create a DNAT entry. After the DNAT entry is created, requests destined for the specified public IP address are forwarded to the ECS instances within a VPC network based on the port mapping rule.

Each DNAT entry consists of the following elements:

- **Public IP address**: the EIP associated with the NAT gateway.
- **Private IP address**: the private IP address assigned to the ECS instance in the VPC network.
- **Public Port**: the external port where requests from the Internet are received.
- **Private Port**: the internal port to which the requests received on the external port are forwarded.
- **Protocol Type**: the protocol used by the ports.

Port mapping and IP mapping

The DNAT feature supports port mapping and IP mapping:

- Port mapping

After port mapping is configured, a NAT gateway forwards requests destined for a public IP address to the specified ECS instance based on the specified protocol and ports.

DNAT entry	Public IP address	Public port	Private IP address	Private port	Protocol type
Entry 1	139.224.xx.xx	80	192.168.x.x	80	TCP
Entry 2	139.224.xx.xx	8080	192.168.x.x	8000	UDP

Entry 1: The NAT gateway forwards requests destined for TCP port 80 of ECS instance 139.244.xx.xx to TCP port 80 of ECS instance 192.168.x.x.

Entry 2: The NAT gateway forwards requests destined for UDP port 8080 of ECS instance 139.224.xx.xx to UDP port 8000 of ECS instance 192.168.x.x.

- IP mapping

  After IP mapping is configured, a NAT gateway forwards all requests destined for a public IP address to the specified ECS instance.

DNAT entry	Public IP address	Public port	Private IP address	Private port	Protocol type
Entry 3	139.224.xx.xx	Any	192.168.x.x	Any	Any

Entry 3: The NAT gateway forwards requests destined for ECS instance 139.224.xx.xx to ECS instance 192.168.x.x.

23.1.6.2. Create a DNAT entry

This topic describes how to create a Destination Network Address Translation (DNAT) entry. DNAT maps a public IP address to an Elastic Compute Service (ECS) instance in a Virtual Private Cloud (VPC) network. This allows the ECS instance to receive requests sent over the Internet. DNAT supports port mapping and IP mapping.

Prerequisites

A NAT gateway is created and associated with an elastic IP address (EIP). For more information, see Create a NAT gateway and Associate an EIP with a NAT Gateway.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateway** page, find the NAT gateway that you want to manage, and click **Configure DNAT** in the **Actions** column.
4. On the **DNAT Table** page, click **Create DNAT Entry**.
5. In the **Create DNAT Entry** panel, set the following parameters and click **OK**.

Parameter	Description
Public IP Address	Select an available public IP address. **Note** If a public IP address is already used to create an SNAT entry, the public IP address cannot be used to create a DNAT entry.

Parameter	Description
Private IP Address	Specify the ECS instance that uses the DNAT entry to communicate with the Internet. You can specify the private IP address of the ECS instance in the following ways: ○ **Auto Fill**: Select the ECS instance from the drop-down list. ○ **Manually Input**: Enter the private IP address of the ECS instance. **Note** This private IP address must fall within the CIDR block of the virtual private cloud (VPC). You can also enter the private IP address of an existing ECS instance.
Port Settings	Select a DNAT mapping method: ○ **All**: This method uses IP mapping. All requests destined for the elastic IP address (EIP) are forwarded to the ECS instance. ○ **Specific Port**: This method uses port mapping. The NAT gateway forwards requests that use the specified protocol and port to the specified port of the ECS instance. After you select Specific Port, specify the **Public Port** (the external port), **Private Port** (the internal port), and **IP Protocol** (the protocol over which data is transferred).
Entry Name	Enter a name for the DNAT entry. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), and hyphens (-). It must start with a letter.

23.1.6.3. Modify a DNAT entry

This topic describes how to modify a Destination Network Address Translation (DNAT) entry. After you create a DNAT entry, you can modify the public IP address, private IP address, ports, and name of the DNAT entry.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateway** page, find the NAT gateway that you want to manage, and click **Configure DNAT** in the **Actions** column.
4. On the **DNAT table** page, find the target DNAT entry, and click **Edit** in the **Actions** column.
5. In the **Edit DNAT Entry** dialog box, change the public IP address, private IP address, ports, and name of the DNAT entry, and then click **OK**.

23.1.6.4. Delete a DNAT entry

This topic describes how to delete a Destination Network Address Translation (DNAT) entry. If you no longer need an Elastic Compute Service (ECS) instance to receive requests sent over the Internet, you can delete the DNAT entry of the ECS instance.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.

3. On the **NAT Gateway** page, find the NAT gateway that you want to manage, and click **Configure DNAT** in the **Actions** column.
4. On the **DNAT table** page, find the target DNAT entry, and click **Remove** in the **Actions** column.
5. In the dialog box that appears, click **OK**.

23.1.7. Manage an SNAT table

23.1.7.1. SNAT table overview

NAT Gateway supports Source Network Address Translation (SNAT). SNAT allows Elastic Compute Service (ECS) instances in a Virtual Private Cloud (VPC) network to access the Internet without using public IP addresses.

SNAT entries

You can create SNAT entries in an SNAT table to allow ECS instances to access the Internet.

An SNAT entry consists of the following elements:

- **VSwitch or ECS instance**: the VSwitch or ECS instance that requires the SNAT proxy service.
- **Public IP address**: the public IP address used to access the Internet.

VSwitch granularity and ECS granularity

SNAT entries can be created based on the following granularity to enable ECS instances in a VPC network to access the Internet.

- VSwitch granularity

  You can select the VSwitch granularity to create an SNAT entry. The NAT gateway provides proxy service for an ECS instance attached to the specified VSwitch by using a specified public IP address when the instance sends requests to the Internet. By default, all ECS instances attached to the VSwitch can use the specified public IP address to access the Internet.

  > **Note** SNAT entries do not take effect on ECS instances that are assigned public IP addresses. For example, an ECS instance may be assigned static public IP address, associated with an elastic IP address (EIP) or has a Destination Network Address Translation (DNAT) IP mapping configured. These ECS instances use the public IP addresses instead of the SNAT entries to access the Internet.

- ECS granularity

  If you select the ECS granularity to create an SNAT entry, the specified ECS instance uses the specified public IP address to access the Internet. The NAT gateway provides proxy service (SNAT) for a specified ECS instance by using a specified public IP address when the instance sends requests to the Internet.

23.1.7.2. Create an SNAT entry

This topic describes how to create a Source Network Address Translation (SNAT) entry. SNAT allows Elastic Compute Service (ECS) instances in a Virtual Private Cloud (VPC) network to access the Internet without using public IP addresses.

Prerequisites

Before you create an SNAT entry, make sure that the following requirements are met:

- A NAT gateway is created and associated with an elastic IP address (EIP). For more information, see Create a NAT gateway and Associate an EIP with a NAT Gateway.
- To create an SNAT entry with VSwitch granularity, make sure that the VSwitch is created and associated with the NAT gateway in a VPC network.
- To create an SNAT entry with ECS granularity, make sure that the ECS instance is created and associated with

the NAT gateway in a VPC network.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateway** page, find the NAT gateway and click **Configure SNAT** in the **Actions** column.
4. On the **SNAT Table** page, click **Create SNAT Entry**.
5. In the **Create SNAT Entry** panel, set the following parameters and click **OK**.

Parameter	Description
VSwitch Granularity	
VSwitch	Select a vSwitch in the VPC. All ECS instances in the vSwitch can access the Internet by using SNAT. **Note** SNAT entries do not take effect on ECS instances that are assigned public IP addresses. For example, an ECS instance may be assigned a static public IP address, associated with an elastic IP address (EIP), or configured with DNAT IP mapping. Such an ECS instance uses the public IP address instead of the SNAT entry to access the Internet.
VSwitch CIDR Block	The CIDR block of the selected vSwitch.
Public IP Address	Select the EIP that is used to access the Internet. You can select one or more EIPs. You can use multiple EIPs to create an SNAT IP address pool. **Note** An EIP that is already used in a DNAT entry cannot be used in an SNAT entry.
ECS Granularity	
Available ECS Instances	Select an ECS instance in the VPC. The ECS instance can access the Internet by using the specified EIP. Make sure that the following requirements are met: ○ The ECS instance is in the Running state. ○ The ECS instance is not assigned an EIP or a static public IP address.
ECS CIDR Block	The CIDR block of the ECS instance.
Public IP Address	Select the EIP that is used to access the Internet. You can select one or more EIPs. You can use multiple EIPs to create an SNAT IP address pool. **Note** An EIP that is already used in a DNAT entry cannot be used in an SNAT entry.

23.1.7.3. Modify an SNAT entry

This topic describes how to modify a Source Network Address Translation (SNAT) entry. After you create an SNAT entry, you can modify the public IP address and name of the SNAT entry.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateway** page, find the NAT gateway and click **Configure SNAT** in the **Actions** column.
4. On the **SNAT table** page, find the target SNAT entry, and click **Edit** in the **Actions** column.
5. In the **Edit SNAT Entry** dialog box, change the public IP address and name of the SNAT entry, and then click **OK**.

23.1.7.4. Delete a SNAT entry

This topic describes how to delete a Source Network Address Translation (SNAT) entry. You can delete the SNAT entry if the Elastic Compute Service (ECS) instances without public IP addresses in a Virtual Private Cloud (VPC) network no longer need the SNAT service to access the Internet.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateway** page, find the NAT gateway and click **Configure SNAT** in the **Actions** column.
4. On the **SNAT table** page, find the target SNAT entry, and click **Remove** in the **Actions** column.
5. In the dialog box that appears, click **OK**.

23.1.8. NAT service plan

23.1.8.1. Create a NAT service plan

You can associate an elastic IP address (EIP) or a NAT service plan to a NAT gateway. However, you can choose only one of them for the NAT gateway. If you want to associate a NAT service plan with the NAT gateway, you must create a NAT service plan first. Then, you can configure SNAT or DNAT for the NAT gateway. A NAT service plan consists of public IP addresses and Internet bandwidth.

Procedure

1. Log on to the NAT Gateway console.
2. On the **NAT Gateways** page, find the target NAT gateway and choose **Purchase NAT Bandwidth Package** in the **Internet Shared Bandwidth** column.
3. On the **Bandwidth Package Details** page, click **Purchase**.
4. On the **NAT Bandwidth Package** page, set the following parameters, and click **Submit**.

Parameter	Description
Region	Indicates the region for which the NAT service plan is purchased.
Billing methods	Select the billing method of the NAT service plan. Only **By Bandwidth** is supported.

Parameter	Description
Bandwidth (Mbit/s)	Enter a bandwidth value for the NAT service plan that you want to purchase. The maximum value is 5000 Mbit/s.
Name	Enter a name for the NAT service plan. The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), and hyphens (-). It must start with a letter or Chinese character.
Description	Enter a description for the NAT service plan. The description must be 2 to 256 characters in length and cannot start with `http://` or `https://`.
Quantity	Enter the number of NAT bandwidth plans that you want to purchase.

23.1.8.2. Modify the bandwidth of a NAT service plan

This topic describes how to modify the bandwidth of a NAT bandwidth plan. The modification takes effect immediately.

Procedure

1. Log on to the NAT Gateway console.
2. On the **NAT Gateways** page, find the target NAT gateway and click the ID of the NAT service plan in the **Internet Shared Bandwidth** column.
3. On the **Bandwidth Package Details** page, click the target NAT service plan, and then choose **Modify Bandwidth**.
4. On the **Modify Bandwidth** page, modify the bandwidth, and then click **Submit**.

   Each NAT bandwidth plan supports a maximum of 5,000 Mbit/s in bandwidth.

23.1.8.3. Add an IP address

This topic describes how to add IP addresses to a NAT service plan. The added IP addresses can be used to create Source Network Address Translation (SNAT) and Destination Network Address Translation (DNAT) rules.

Procedure

1. Log on to the NAT Gateway console.
2. On the **NAT Gateways** page, find the target NAT gateway and click the ID of the NAT service plan in the **Internet Shared Bandwidth** column.
3. On the **Bandwidth Package Details** page, click the target NAT service plan, and then choose **Add IP Address**.
4. On the **Modify IP Addresses** page, enter the number of IP addresses to be added, and then click **Submit**.

23.1.8.4. Release an IP address

This topic describes how to release IP addresses in a NAT service plan. The NAT service plan must contain at least one IP address.

Prerequisites

Before you release an IP address in the NAT service plan, make sure that the IP address is not used in Source

Network Address Translation (SNAT) and Destination Network Address Translation (DNAT) entries. If the IP address is used in an SNAT or DNAT entry, delete the SNAT or DNAT entry first. For more information, see Delete a DNAT entry and Delete a SNAT entry.

Procedure

1. Log on to the NAT Gateway console.
2. On the **NAT Gateways** page, find the target NAT gateway and click the ID of the NAT service plan in the **Internet Shared Bandwidth** column.
3. On the **Bandwidth Package Details** page, click the target NAT service plan.
4. In the **Public IP List** section, find the target IP address, and click **Release** in the **Actions** column.
5. In the **Release IP** dialog box, click **OK**.

23.1.8.5. Delete a NAT service plan

This topic describes how to delete a service plan.

Prerequisites

Before you start, make sure that the following requirements are met:

- Delete the IP addresses that are used in Destination Network Address Translation (DNAT) entries. For more information, see Delete a DNAT entry.
- Delete the IP addresses that are used for Source Network Address Translation (SNAT) entries. For more information, see Delete a SNAT entry.

Procedure

1. Log on to the NAT Gateway console.
2. On the **NAT Gateways** page, find the target NAT gateway and click the ID of the NAT service plan in the **Internet Shared Bandwidth** column.
3. On the **Bandwidth Package Details** page, find the target NAT service plan and click **Delete**.
4. In the **Delete Shared Internet Shared Bandwidth** dialog box, click **OK**.

23.1.9. Anti-DDoS Basic

Distributed Denial of Service (DDoS) attack is a malicious network attack against the target system, which can make the attacked network inaccessible. Alibaba Cloud provides up to 5 Gbit/s of basic anti-DDoS protection for NAT Gateway, which can efficiently prevent DDoS attack.

How Anti-DDoS Basic works

After you enable Anti-DDos Basic, all traffic from the Internet must first pass through Alibaba Cloud Security before arriving at NAT Gateway. Anti-DDoS Basic scrubs and filters common DDoS attacks at Alibaba Cloud Security. Anti-DDos Basic protects your services against attacks such as SYN flood, UDP flood, ACK flood, ICMP flood, and DNS Query flood.

Anti-DDoS Basic sets the scrubbing threshold and black hole triggering threshold based on the EIP bandwidth of NAT Gateway. When the inbound traffic reaches the threshold, scrubbing or blackholing is triggered:

- Scrubbing: When the attack traffic from the Internet exceeds the scrubbing threshold or matches certain attack traffic pattern, Alibaba Cloud Security starts scrubbing the attack traffic. The scrubbing includes packet filtering, bandwidth capping, and traffic throttling.
- Blackholing: When the attack traffic from the Internet exceeds the black hole triggering threshold, blackholing is triggered and all inbound traffic is dropped.

Scrubbing threshold

The thresholds for triggering traffic scrubbing and blackholing on NAT Gateway are calculated as described in the following table:

EIP bandwidth	Traffic scrubbing threshold (bits/s)	Traffic scrubbing threshold (packets/s)	Default black hole triggering threshold
Lower than or equal to 800 Mbit/s	800Mbps	120,000	1.5 Gbps
Higher than 800 Mbit/s	Predefined bandwidth	Predefined bandwidth × 150	Predefined bandwidth × 2

If the EIP bandwidth is 1,000 Mbit/s, the traffic scrubbing threshold (bits/s) is 1,000 Mbit/s, the traffic scrubbing threshold (packets/s) is 150,000 and the default blackholing threshold is 2 Gbit/s.

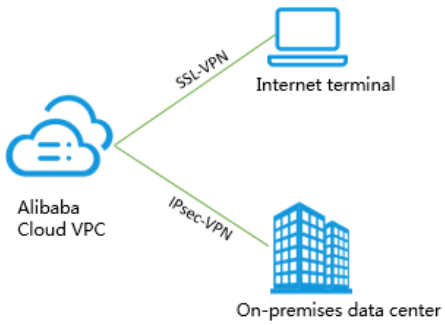
24.VPN Gateway

24.1. User Guide

24.1.1. What is VPN Gateway?

VPN Gateway is an Internet-based service that allows you to connect enterprise data centers, office networks, or Internet-facing terminals to Alibaba Cloud Virtual Private Cloud (VPC) networks through secure and reliable connections. VPN Gateway supports both IPsec-VPN connections and SSL-VPN connections.

> **Note** The Alibaba Cloud VPN Gateway service complies with the local regulations and policies. VPN Gateway does not provide Internet access services.



Features

VPN Gateway supports the following features:

- IPsec-VPN

  Route-based IPsec-VPN allows you to route network traffic in multiple ways, and also facilitates the configuration and maintenance of VPN policies.

  You can use IPsec-VPN to connect an on-premises data center to a VPC network or connect two VPC networks. IPsec-VPN supports the IKEv1 and IKEv2 protocols. Any devices that support these two protocols can connect to Alibaba Cloud VPN Gateway, such as devices manufactured by Huawei, H3C, Hillstone, Sangfor, Cisco ASA, Juniper, SonicWall, Nokia, IBM, and Ixia.

- SSL-VPN

  SSL-VPN is implemented based on the OpenVPN framework. You can create an SSL-VPN connection to connect a remote client to applications and services deployed in a VPC network. After you deploy your applications or services, you only need to import the certificate to the client to initiate a connection.

Benefits

VPN Gateway offers the following benefits:

- High security: You can use the IKE and IPsec protocols to encrypt data for secure and reliable data transmission.
- High availability: VPN Gateway adopts the hot-standby architecture to achieve failover within a few seconds, session persistence, and zero service downtime.
- Cost-effectiveness: The encrypted Internet connections provided by VPN Gateway are more cost-effective than leased lines.
- Ease of use: VPN Gateway is a ready-to-use service. VPN gateways start to work immediately after they are deployed.

24.1.2. Log on to the VPN Gateway console

This topic describes how to log on to the Apsara Uni-manager Management Console. You can manage your VPN gateways in the console. The Google Chrome browser is used as an example.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Login**.
4. In the top menu bar, choose **Products > Networking > Virtual Private Cloud**.
5. In the left-side navigation pane, choose **Interconnections > VPN**.

24.1.3. Get started with IPsec-VPN

24.1.3.1. Connect on-premises data centers to VPC networks

This topic describes how to create IPsec-VPN connections on VPN gateways to connect an on-premises data center to a Virtual Private Cloud (VPC) network.

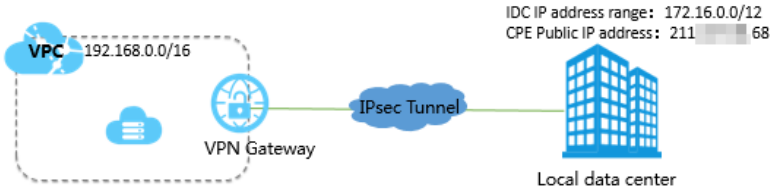
Prerequisites

Before you start, make sure that the following requirements are met:

- Check the gateway device in the on-premises data center. Alibaba Cloud VPN gateways support the standard IKEv1 and IKEv2 protocols. Any gateway device that supports these two protocols can connect to Alibaba Cloud VPN gateways, such as gateway devices manufactured by H3C,Huawei, Hillstone, Sangfor, Cisco ASA, Juniper, SonicWall, Nokia, IBM, and Ixia.
- Make sure that you have set a static public IP address for the gateway device in the on-premises data center.
- The CIDR block of the on-premises data center must not overlap with that of the VPC network.

Context

For example, a company creates a VPC network on Alibaba Cloud. The CIDR block of the VPC network is 192.168.0.0/16. The CIDR block of the on-premises data center is 172.16.0.0/12. The static public IP address for the gateway device in the on-premises data center is 211.xx.xx.68. To meet business requirements, the company needs to connect the on-premises data center to the VPC network.



The preceding figure displays that the on-premises data center is connected to the VPC network through IPsec-VPN. Cloud resources can be shared with on-premises data centers.

Step 1: Create a VPN gateway

Take the following steps to create a VPN gateway:

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose **VPN > VPN Gateways**.
3. On the **VPN Gateways** page, click **Create VPN Gateway**.
4. On the **Create VPN Gateway** page, set the following parameters for the VPN gateway, and then click **Submit**.
   - **Organization**: Select the organization to which the VPN gateway belongs.
   - **Resource Set**: Select the resource set to which the VPN gateway belongs.
   - **Region**: Select the region where you want to deploy the VPN gateway.

     > **Note** Make sure that the VPC network and the VPN gateway associated with the VPC network are deployed in the same region.

   - **Name**: Enter a name for the VPN gateway.

     The name must be 2 to 128 characters in length, and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It must start with a letter or Chinese character and cannot start with `http://` or `https://`.
   - **VPC**: Select the VPC network to be associated with the VPN gateway.
   - **Bandwidth**: Specify the maximum bandwidth of the VPN gateway. The bandwidth is provided for data transfer over the Internet.
   - **IPsec-VPN**: Specify whether to enable IPsec-VPN for the VPN gateway. In this example, select **Enable**.

     After IPsec-VPN is enabled, you can create IPsec-VPN connections between an on-premises data center and a VPC network, or between two VPC networks.
   - **SSL-VPN**: Specify whether to enable SSL-VPN. In this example, select **Disable**.

     SSL-VPN connections are point-to-site connections. SSL-VPN allows you to connect a client to Alibaba Cloud without configuring a customer gateway.
   - **SSL Connections**: Specify the maximum number of concurrent SSL connections that the VPN gateway supports.

     > **Note** This parameter is available only after SSL-VPN is enabled.

5. Go to the VPN Gateways page to view the newly created VPN gateway.

   The newly created VPN gateway is in the Preparing state. Its status changes to Normal after about two

minutes. The Normal state indicates that the VPN gateway is initialized and ready for use.

> **Note** It takes about one to five minutes to create a VPN gateway.

Step 2: Create a customer gateway

Take the following steps to create a customer gateway.

1. In the left-side navigation pane, choose **VPN > Customer Gateways**.
2. Select the region where you want to deploy the customer gateway
3. On the **Customer Gateways** page, click **Create Customer Gateway**.
4. On the **Create Customer Gateway** page, set the following parameters, and click **Submit**.
   - **Organization**: Select the organization to which the customer gateway belongs.
   - **Resource Set**: Select the resource set to which the customer gateway belongs.
   - **Region**: Select the region where you want to deploy the customer gateway.

     > **Note** Make sure that the customer gateway and the VPN gateway to be connected are deployed in the same region.

   - **Zone**: Select the zone where you want to deploy the customer gateway.
   - **Name**: Enter a name for the customer gateway.

     The name must be 2 to 128 characters in length, and can contain letters, digits, hyphens (-), and underscores (_). It must start with a letter or Chinese character and cannot start with `http://` or `https://`.
   - **IP Address**: Enter the public IP address of the gateway device in the on-premises data center that is to be connected to the VPC network. In this example, enter **211.xx.xx.68**.
   - **Description**: Enter a description for the customer gateway.

     The description must be 2 to 256 characters in length, and cannot start with `http://` or `https://`.

Step 3: Create an IPsec-VPN connection

Take the following steps to create an IPsec-VPN connection:

1. In the left-side navigation pane, choose **VPN > IPsec Connections**.
2. Select the region where you want to create an IPsec-VPN connection.
3. On the **IPsec Connections** page, click **Create IPsec Connection**.
4. On the **Create IPsec Connection** page, set the following parameters for the IPsec-VPN connection, and click **Submit**.
   - **Organization**: Select the organization to which the IPsec-VPN connection belongs.
   - **Resource Set**: Select the resource set to which the IPsec-VPN connection belongs.
   - **Region**: Select the region where the IPsec-VPN connection is established.
   - **Zone**: Select the zone where the IPsec-VPN connection is established.
   - **Name**: Enter a name for the IPsec-VPN connection.
   - **VPN Gateway**: Select a VPN gateway.
   - **Customer Gateway**: Select the customer gateway to be connected through the IPsec-VPN connection.
   - **Source CIDR Block**: Enter the CIDR block of the VPC network with which the selected VPN gateway is associated. In this example, enter **192.168.0.0/16**.
   - **Destination CIDR Block**: Enter the CIDR block of the on-premises data center. In this example, enter

172.16.0.0/12.

- **Immediate Effect**: Specify whether to start connection negotiations immediately.
  - **Yes**: negotiate immediately after the configuration is complete.
  - **No**: negotiate when traffic is detected in the IPsec-VPN connection.
- **Pre-shared Key**: Enter the pre-shared key. The pre-shared key must be the same as that of the gateway device deployed in the on-premises data center.

  Use the default settings for other parameters.

Step 4: Load the configurations of the IPsec-VPN connection to the customer gateway device

Take the following steps to load the configurations of the IPsec-VPN connection to the customer gateway device:

1. In the left-side navigation pane, choose **VPN > IPsec Connections**.
2. Select the region where the IPsec-VPN connection is established.
3. On the **IPsec Connections** page, find the target IPsec-VPN connection, and then choose **More > Download Configuration** in the **Actions** column.
4. Load the configurations of the IPsec-VPN connection to the customer gateway device by following the instructions described in . For more information about how to configure customer gateways, consult the manufacturers of the gateway devices.

   RemotSubnet and LocalSubnet in the downloaded configurations are opposite to RemotSubnet and LocalSubnet that you specify when you create an IPsec-VPN connection. For a VPN gateway, RemotSubnet refers to the CIDR block of the on-premises data center and LocalSubnet refers to the CIDR block of the VPC network. For a customer gateway, LocalSubnet refers to the CIDR block of the on-premises data center and RemoteSubnet refers to the CIDR block of the VPC network.

Step 5: Configure routes for the VPN gateway

Take the following steps to configure routes for the VPN gateway:

1. In the left-side navigation pane, choose **VPN > VPN Gateways**.
2. Select the region where the VPN gateway is deployed.
3. On the **VPN Gateways** page, find the target VPN gateway, and then click the instance ID in the **Instance ID/Name** column.
4. In the **Destination-based routing** tab, click **Add Route Entry**.
5. In the **Add Route Entry** dialog box, set the following parameters and click **OK**.
   - **Destination CIDR Block**: Enter the CIDR block of the on-premises data center. In this example, enter **172.16.0.0/12**.
   - **Next Hop Type**: Select IPsec Connection.
   - **Next Hop**: Select an IPsec instance.
   - **Publish to VPC**: Specify whether to automatically publish new route entries to the VPC route table. In this example, select **Yes**.
   - **Weight**: Select a weight. In this example, select **100**.

Step 6: Verify the settings

Log on to an Elastic Compute Service (ECS) instance that is not assigned a public IP address in the VPC network. Run the **ping** command to **ping** the private IP address of a server that resides in the on-premises data center, and test the connectivity.

24.1.4. Get started with SSL-VPN

24.1.4.1. Initiate a connection from a Linux client

This topic describes how to use SSL-VPN to connect a Linux client to a Virtual Private Cloud (VPC) network.

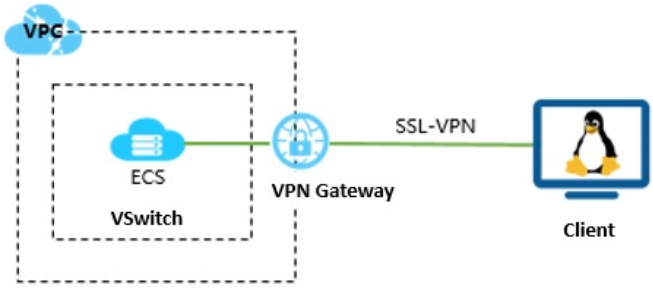
Prerequisites

Before you start, make sure that the following requirements are met:

- The CIDR block of the VPC network must not overlap with that of the client. Otherwise, the client cannot communicate with the VPC network.
- The client must be able to access the Internet.

Context

The following scenario is used as an example.



Step 1: Create a VPN gateway

Take the following steps to create a VPN gateway.

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose **VPN > VPN Gateways**.
3. On the **VPN Gateways** page, click **Create VPN Gateway**.
4. On the **Create VPN Gateway** page, specify the following parameters for the VPN gateway, and then click **Submit**.
   - **Organization**: Select the organization to which the VPN gateway belongs.
   - **Resource Set**: Select the resource set to which the VPN gateway belongs.
   - **Region**: Select the region where the VPN gateway is deployed.

     > **Note** Make sure that the VPC network and the VPN gateway associated with the VPC network are deployed in the same region.

   - **Instance Name**: Enter a name for the VPN gateway.

     The name must be 2 to 128 characters in length, and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It must start with a letter or Chinese character, and cannot start with `http://` or `https://`.
   - **VPC**: Select the VPC network to be associated with the VPN gateway.
   - **Bandwidth**: Specify the maximum bandwidth of the VPN gateway. The bandwidth is provided for data transfer over the Internet.

- **IPsec-VPN**: Specify whether to enable IPsec-VPN for the VPN gateway. In this example, select **Disable**.
- **SSL-VPN**: Specify whether to enable SSL-VPN for the VPN gateway. In this example, select **Enable**.

  SSL-VPN connections are point-to-site connections. SSL-VPN allows you to connect a client to Alibaba Cloud without configuring a gateway for the client.
- **SSL Connections**: Specify the maximum number of concurrent SSL connections that the VPN gateway supports.

5. Go to the VPN Gateways page to view the newly created VPN gateway.

   The newly created VPN gateway is in the Preparing state. Its status changes to Normal after about two minutes. The Normal status indicates that the VPN gateway is initialized and ready for use.

   > **Note** It takes about one to five minutes to create a VPN gateway.

Step 2: Create an SSL server

Take the following steps to create an SSL server.

1. In the left-side navigation pane, choose **VPN > SSL Servers**.
2. Select the region where you want to create the SSL server.
3. On the **SSL Servers** page, click **Create SSL Server**.
4. On the **Create SSL Server** page, set the following parameters for the SSL server, and then click **Submit**.
   - **Organization**: Select the organization to which the SSL server belongs.
   - **Resource Set**: Select the resource set to which the SSL server belongs.
   - **Region**: Select the region where you want to deploy the SSL server.
   - **Zone**: Select the zone where you want to deploy the SSL server.
   - **Name**: Enter a name for the SSL server.
   - **VPN Gateway**: Select a VPN gateway from the drop-down list.
   - **Source CIDR Block**: Enter the CIDR block of the VPC network. Click **+Add Local Network** to add more CIDR blocks. You can add the CIDR block of a VPC network, a VSwitch, and a local network.
   - **Client CIDR Block**: Enter the CIDR block of the client. The client connects to the SSL server from the specified CIDR block.
   - **Advanced Settings**: Specify whether to customize the advanced settings. In this topic, the default settings are used.

Step 3: Create and download an SSL client certificate

Take the following steps to create and download an SSL client certificate.

1. In the left-side navigation pane, choose **VPN > SSL Clients**.
2. Select the region where the client is created.
3. On the **SSL Clients** page, click **Create Client Certificate**.
4. On the **Create SSL Client Certificate** page, set the following parameters for the SSL client, and then click **Submit**.
   - **Organization**: Select the organization to which the SSL client certificate belongs.
   - **Resource Set**: Select the resource set to which the SSL client certificate belongs.
   - **Region**: Select the region where you want to create the SSL client certificate.
   - **Zone**: Select the zone where you want to create the SSL client certificate.
   - **Name**: Enter a name for the SSL client certificate.
   - **VPN Gateway**: Select the VPN gateway to be associated with the SSL client certificate.

- **SSL Server**: Select the SSL server to which you want to import the SSL client certificate.

5. On the **SSL Clients** page, find the target SSL client certificate, and then click **Download** in the **Actions** column.

Step 4: Configure the client

Take the following steps to configure the Linux client:

1. Run the following command to install the OpenVPN:

```
yum install -y openvpn
```

2. Decompress the client certificate package that you downloaded in Step 3 and copy the client certificate file to the */etc/openvpn/conf/* folder where OpenVPN is installed.
3. Run the following command to launch OpenVPN:

```
openvpn --config /etc/openvpn/conf/config.ovpn --daemon
```

Step 5: Test the connectivity

To test the connectivity, run the **ping** command to ping the connected ECS instance in the VPC network.

> **Note** Make sure that the security group rules of the ECS instance allow remote access from Linux clients.

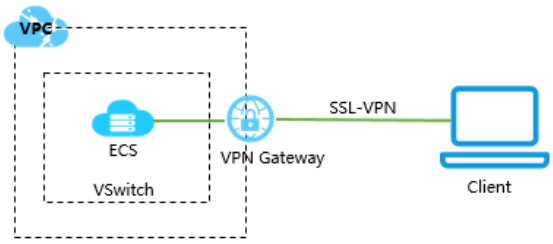
24.1.4.2. Initiate a connection from a Windows client

This topic describes how to use SSL-VPN to connect a Windows client to a Virtual Private Cloud (VPC) network.

Prerequisites

Context

The following scenario is used as an example.

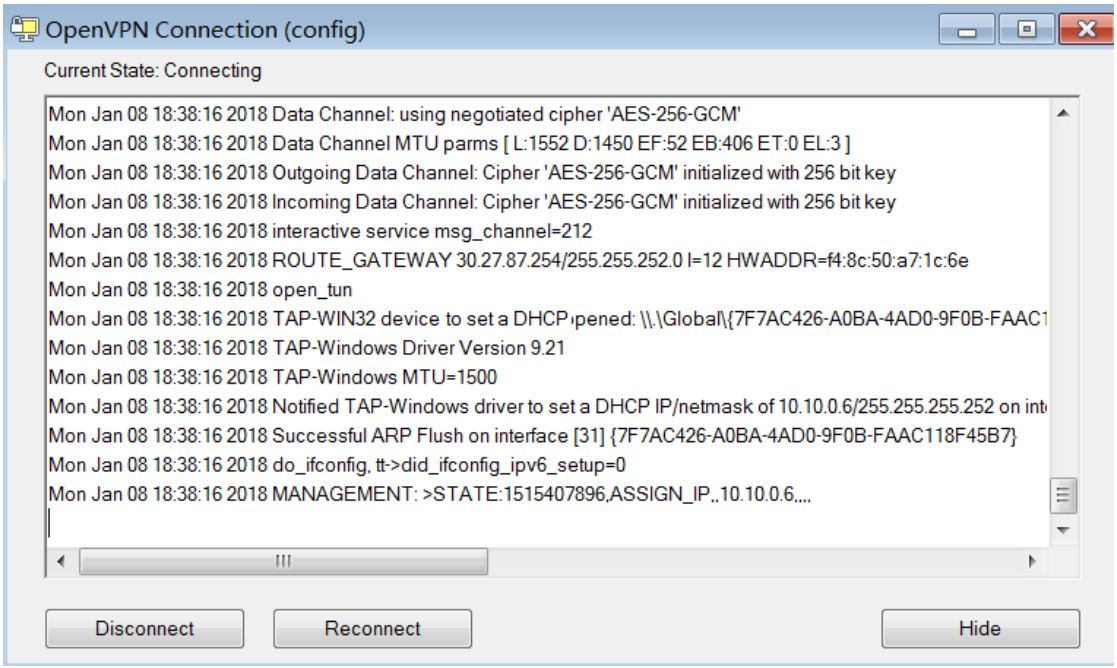


Step 4: Configure the client

Take the following steps to configure the Windows client:

> **Notice** You must run the client as an administrator.

1. Download and install OpenVPN.
2. Decompress the client certificate package that you downloaded in Step 3 and copy the client certificate file to the *config* folder where OpenVPN is installed.
3. Click **Connect** to initiate a connection.



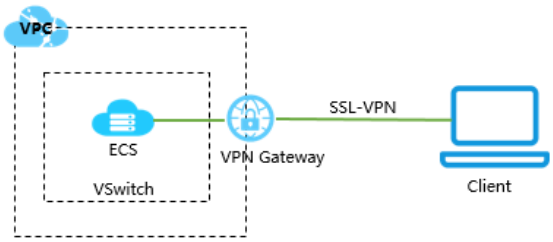
24.1.4.3. Initiate a connection from a macOS client

This topic describes how to use SSL-VPN to connect a macOS client to a Virtual Private Cloud (VPC) network.

Prerequisites

Context

The following scenario is used as an example.



Step 4: Configure the client

Take the following steps to configure the macOS client:

1. Run the following command to install OpenVPN:

   ```
   brew install openvpn
   ```

   > **Note** If Homebrew is not installed, install Homebrew first.

2. Decompress the client certificate package that you downloaded in Step 3, copy the client certificate file to the folder where OpenVPN is installed, and initiate a connection.
   i. Back up the default configuration file.
   ii. Run the following command to delete the default configuration file:

   ```
   rm /usr/local/etc/openvpn/*
   ```

iii. Run the following command to copy the file to the configuration directory:

```
cp cert_location /usr/local/etc/openvpn/
```

In the preceding command, `cert_location` represents the path that stores the certificate downloaded in Step 3, for example, */Users/example/Downloads/certs6.zip*.

iv. Run the following command to decompress the client certificate package:

```
cd  /usr/local/certificates
unzip /usr/local/etc/openvpn/certs6.zip
```

v. Run the following command to initiate a connection:

```
sudo /usr/local/opt/openvpn/sbin/openvpn --config /usr/local/etc/openvpn/config.ovpn
```

24.1.5. Manage a VPN Gateway

24.1.5.1. Create a VPN gateway

This topic describes how to create a VPN gateway. You must create a VPN gateway before you can use the IPsec-VPN and SSL-VPN services. After the VPN gateway is created, a public IP address is assigned to the VPN gateway.

Procedure

1. Log on to the VPN Gateway console.
2.
3. On the **VPN Gateways** page, click **Create VPN Gateway**.
4. On the **Create VPN** page, specify the following parameters for the VPN gateway, and then click **Submit**.

Parameter	Description
Organization	Select the organization to which the VPN gateway belongs.
Resource Set	Select the resource set to which the VPN gateway belongs.
Region	Select the region where the VPN gateway is deployed. You can use IPsec-VPN to connect an on-premises data center to a VPC network or connect two VPC networks. Make sure that the VPC network and the VPN gateway associated with the VPC network are deployed in the same region.
Instance Name	Enter a name for the VPN gateway. The name must be 2 to 128 characters in length, and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It must start with a letter or Chinese character and cannot start with `http://` or `https://`.
VPC	Select the VPC network to be associated with the VPN gateway.
Bandwidth	Specify the maximum bandwidth of the VPN gateway. The bandwidth is provided for data transfer over the Internet.
IPsec-VPN	Specify whether to enable IPsec-VPN for the VPN gateway. After IPsec-VPN is enabled, you can create IPsec-VPN connections between an on-premises data center and a VPC network, or between two VPC networks.

Parameter	Description
SSL-VPN	Specify whether to enable SSL-VPN for the VPN gateway. SSL-VPN connections are point-to-site connections. SSL-VPN allows you to connect a client to Alibaba Cloud without configuring a customer gateway.
SSL Connections	Specify the maximum number of concurrent SSL connections that the VPN gateway supports. **Note** This parameter is available only after SSL-VPN is enabled.

24.1.5.2. Modify a VPN gateway

This topic describes how to modify the name and description of a VPN gateway.

Prerequisites

A VPN gateway is created. For more information, see Create and manage a VPN gateway.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose **VPN > VPN Gateways**.
3. Select the region where the VPN gateway is deployed.
4. On the **VPN Gateways** page, find the target VPN gateway, and click the icon in the **Instance ID/Name** column. In the dialog box that appears, enter a new name and click **OK**.

   The name must be 2 to 100 characters in length, and can contain letters, digits, underscores (_), and hyphens (-). It must start with a letter or Chinese character.
5. Click the icon in the **Description** column. In the dialog box that appears, enter a new description and click **OK**.

   The description must be 2 to 256 characters in length, and cannot start with http:// or https://.

24.1.5.3. Configure routes of a VPN Gateway

24.1.5.3.1. VPN Gateway route overview

After you create an IPsec-VPN connection, you must manually add a VPN Gateway route.

The route-based IPsec-VPN enables you to easily configure and maintain VPN policies, and provides flexible ways for routing traffic.

You can add the following two types of routes for a VPN Gateway:

- Policy-based routes.
- Destination-based routes.

Policy-based route

If a policy-based route is used, traffic is forwarded based on both the source IP address and the destination IP address.

For more information, see Add a policy-based route entry.

> **Note** Policy-based routes take precedence over destination-based routes.

Destination-based route

If a destination-based route is used, traffic is forwarded based only on the destination IP address.

For more information, see Add a destination-based route entry.

24.1.5.3.2. Add a policy-based route entry

This topic describes how to add a policy-based route entry after an IPsec-VPN connection is created. Policy-based routing (PBR) is a technique that routes packets based on source and destination IP addresses.

Procedure

1. Log on to the VPN Gateway console.
2.
3.
4. On the **VPN Gateways** page, find the target VPN gateway and click the instance ID in the **Instance ID/Name** column.
5. Click the **Policy-based Routing** tab, and then click **Add Route Entry**.
6. In the **Add Route Entry** dialog box, set the following parameters and click **OK**.

Parameter	Description
Destination CIDR Block	The CIDR block that you want to access.
Source CIDR Block	The CIDR block of the VPC network.
Next Hop Type	Select IPsec Connection.
Next Hop	Select an IPsec instance to create an IPsec-VPN connection.
Publish to VPC	Specify whether to automatically publish new route entries to the VPC route table. ○ Yes (Recommended): automatically publishes new route entries to the VPC route table. ○ No: does not automatically publish new route entries to the VPC route table. **Note** If you select No, you must manually publish new route entries to the VPC route table.
Weight	Select a weight. Valid values: ○ 100: indicates that the priority of the route entry is high. ○ 0: indicates that the priority of the route entry is low. **Note** If two policy-based route entries are configured with the same destination CIDR block, you cannot set the weights of both route entries to 100.

24.1.5.3.3. Add a destination-based route entry

This topic describes how to manually add a destination-based route entry after an IPsec-VPN connection is created. Destination-based routing is a technique that routes packets to specified destination IP addresses.

Procedure

1. Log on to the VPN Gateway console.
2.
3.
4.
5. In the **Destination-based routing** tab, click **Add Route Entry**.
6. In the **Add Route Entry** dialog box, set the following parameters and click **OK**.

Parameter	Description
Destination CIDR Block	The CIDR block that you want to access.
Next Hop Type	Select IPsec Connection.
Next Hop	Select an IPsec instance to create an IPsec-VPN connection.
Publish to VPC	Specify whether to automatically publish new route entries to the VPC route table. ○ Yes (Recommended): automatically publishes new route entries to the VPC route table. ○ No: does not automatically publish new route entries to the VPC route table. **Note** If you select No, you must manually publish new route entries to the VPC route table.
Weight	Select a weight. Valid values: ○ 100: indicates that the priority of the route entry is high. ○ 0: indicates that the priority of the route entry is low. **Note** If two destination-based route entries are configure with the same destination CIDR block, you cannot set the weights of both route entries to 100.

24.1.5.4. Delete a VPN gateway

This topic describes how to delete a VPN gateway. After you delete a VPN gateway, you can no longer use the VPN gateway to establish IPsec-VPN or SSL-VPN connections.

Context

Before you delete a VPN gateway, make sure that the following conditions are met:

- The IPsec-VPN connections on the VPN gateway are deleted. For more information, see and Delete an IPsec-VPN connection.
- The SSL server associated with the VPN gateway is deleted. For more information, see and Delete an SSL server.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose **VPN > VPN Gateways**.
3. Select the region where the VPN gateway is deployed.
4. On the **VPN Gateways** page, find the target VPN gateway, and then click **Delete** in the **Actions** column.
5. In the **Delete VPN Gateway** dialog box, click **OK**.

24.1.6. Manage a customer gateway

24.1.6.1. Create a customer gateway

This topic describes how to create a customer gateway when you use IPsec-VPN to connect a Virtual Private Cloud (VPC) network to an on-premises data center or connect two VPC networks. You can register and update the information of an on-premises gateway to Alibaba Cloud by creating a customer gateway, and then connect the customer gateway to the VPN gateway. A customer gateway can be connected to multiple VPN gateways.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, click **VPN > Customer Gateways**.
3. Select the region where you want to deploy the customer gateway.

   > **Note** Make sure that the customer gateway and the VPN gateway to be connected are deployed in the same region.

4. On the **Customer Gateways** page, click **Create Customer Gateway**.
5. On the **Create Customer Gateway** page, set the following parameters, and click **Submit**.

Parameter	Description
Organization	Select the organization to which the customer gateway belongs.
Resource Set	Select the resource set to which the customer gateway belongs.
Region	Select the region where you want to deploy the customer gateway. **Note** Make sure that the customer gateway and the VPN gateway to be connected are deployed in the same region.
Zone	Select the zone where you want to deploy the customer gateway
Name	Enter a name for the customer gateway. The name must be 2 to 128 characters in length, and can contain letters, digits, hyphens (-), and underscores (_). It must start with a letter or Chinese character and cannot start with `http://` or `https://`.
IP Address	Enter the static public IP address of the gateway device that is deployed in the on-premises data center.
Description	Enter a description for the customer gateway. The description must be 2 to 256 characters in length, and cannot start with `http://` or `https://`.

24.1.6.2. Modify a customer gateway

This topic describes how to modify the name and description of a customer gateway.

Prerequisites

A customer gateway is created. For more information, see Create a customer gateway.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose **VPN > Customer Gateways**.
3. Select the region where the customer gateway is deployed.
4. On the **Customer Gateways** page, find the target customer gateway, click the icon in the **Instance ID** column. In the dialog box that appears, enter a name and click **OK**.

   The name must be 2 to 128 characters in length, and can contain digits, underscores (_), and hyphens (-). It must start with a letter or Chinese character.
5. Click the icon in the **Description** column. In the dialog box that appears, enter a new description and click **OK**.

   The description must be 2 to 256 characters in length, and cannot start with http:// or https://.

24.1.6.3. Delete a customer gateway

This topic describes how to delete a customer gateway.

Procedure

1. Log on to the VPN Gateway console.
2.
3.
4. On the **Customer Gateways** page, find the target customer gateway, and then click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

24.1.7. Configure SSL-VPN

24.1.7.1. Configuration overview

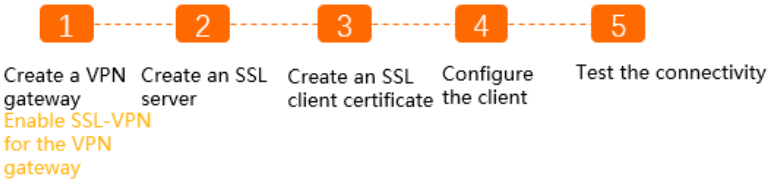
This topic describes how to use the SSL-VPN function to connect a remote client to a VPC.

Prerequisites

Before you use SSL-VPN to establish a connection between a client and a VPC, make sure that the following requirements are met:

- The private CIDR block of the client does not overlap with the private CIDR block of the VPC. Otherwise, the client and the VPC cannot communicate with each other.
- The client can access the Internet.
- You have read and understand the security group rules that apply to the Elastic Compute Service (ECS) instances in the VPC, and make sure that the security rules allow the client to access the ECS instances.

Procedure



1. Create a VPN gateway.

   Create a VPN gateway and enable the SSL-VPN feature.

2. Create an SSL server.

   On the SSL server, specify the private CIDR block that the client needs to access and the CIDR block that is used by the client.

3. Create an SSL client certificate.

   Create and download a client certificate based on the SSL server configuration.

4. Configure the client.

   Download and install VPN software on the client, load the SSL client certificate, and then initiate an SSL-VPN connection.

5. Verify the connectivity.

   Open the CLI on the client, and run the **ping** command to ping an ECS instance in the VPC.

24.1.7.2. Manage an SSL server

24.1.7.2.1. Create an SSL server

This topic describes how to create an SSL server. Before you use SSL-VPN to establish a point-to-site connection, you must create an SSL server.

Prerequisites

A VPN gateway is created and SSL-VPN is enabled for the VPN gateway. For more information, see Create a VPN gateway.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose **Cross-Network Connections > SSL Servers**.
3. In the top navigation bar, select the region where you want to create the SSL server.
4. On the **SSL Servers** page, click **Create SSL Server**.
5. On the **Create SSL Server** page, configure the SSL server based on the following information and click **Submit**.

Parameter	Description
Organization	Select the organization to which the SSL server belongs.
Resource Set	Select the resource set to which the SSL server belongs.
Region	Select the region where the SSL server is to be deployed.
Zone	Select the zone where the SSL server is to be deployed.

Parameter	Description
Name	Enter a name for the SSL server. The name must be 2 to 128 characters in length, and can contain letters, digits, hyphens (-), and underscores (_). It must start with a letter or Chinese character and cannot start with `http://` or `https://`.
VPN Gateway	Select the VPN gateway that you want to associate with the SSL server. Make sure that SSL-VPN is enabled for the VPN gateway.
Source CIDR Block	Enter the CIDR block of the network to be connected to the client through an SSL-VPN connection. It can be the CIDR block of a VPC, a vSwitch, a data center connected to a VPC through a leased line, or a cloud service such as ApsaraDB RDS or Object Storage Service (OSS). Click **+** to add more CIDR blocks. **Note** The subnet mask of the specified server CIDR block must be 16 to 29 bits in length.
Client CIDR Block	Enter the CIDR block to be allocated to the virtual network interface of the client. Do not enter the CIDR block where the client resides. When a client accesses the server through an SSL-VPN connection, the VPN gateway assigns an IP address from the specified client CIDR block to the client. **Note** Make sure that the server CIDR block and the client CIDR block do not overlap with each other.
Advanced Settings	Specify whether to customize the advanced settings. ○ **Default**: Use the default advanced settings. ○ **Custom**: Use custom settings. You can customize the following settings: ■ **Protocol**: Select the protocol over which the SSL-VPN connection is established. Supported protocols are UDP and TCP. ■ **Port**: Specify the port to which the SSL-VPN connection is established. The following ports are not supported: 22, 2222, 22222, 9000, 9001, 9002, 7505, 80, 443, 53, 68, 123, 4510, 4560, 500, and 4500. ■ **Encryption Algorithm**: Select the encryption algorithm used by the SSL connection. Valid values: AES-128-CBC, AES-192-CBC, AES-256-CBC, and none. ■ **Compressed**: Specify whether to enable data compression.

24.1.7.2.2. Modify an SSL server

This topic describes how to modify the name, server CIDR block, client CIDR block, and advanced settings of an SSL server.

Prerequisites

An SSL server is created. For more information, see Create an SSL server.

Procedure

1. Log on to the VPN Gateway console.
2.
3.
4. On the **SSL Servers** page, find the target SSL server, and then click **Edit** in the **Actions** column.
5. On the **Edit SSL Server** page, modify the name, server CIDR block, client CIDR block, and advanced settings of the SSL server, and then click **OK**.

24.1.7.2.3. Configure a routing group

This topic describes how to configure a routing group to control the inbound and outbound traffic of an Elastic Compute Service (ECS) instance after you create an IPsec-VPN connection.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose **VPN > SSL Servers**.
3. Select the region where the SSL server is deployed.
4. On the **SSL Servers** page, find the target SSL server, and then click **Configure Routing Group** in the **Actions** column.
5. In the **Configure Routing Group** panel, set the following parameters and click **OK**.

Parameter	Description
Security Group	Select the security group to which you want to add the security group rule.
Rule Direction	Select the direction of data transfer that the rule controls. ○ **Outbound**: controls data transfer from the ECS instances in the security group to the Internet or other ECS instances. ○ **Inbound**: controls data transfer from the Internet or other ECS instances to the ECS instances in the security group.
Action	Specify the action to be performed on the matching requests. ○ **Allow**: accepts requests. ○ **Deny**: drops requests without returning messages. If two security group rules use the same settings except for different actions, the **Deny** rule prevails over the **Allow** rule.
Protocol Type	The protocol of the security group rule.
Port Range	Enter a port range for the security group rule. Valid values: -1 and 1 to 65535. You cannot enter only -1. Examples: ○ 1/200 specifies ports 1 to 200. ○ 80/80 specifies port 80. ○ -1/-1 specifies all ports.
Priority	Set the priority of the rule. Valid values: 1 to 100. The default value is 1, which indicates the highest priority.

Parameter	Description
Authorization Mode	Specify the type of addresses that the security group rule permits or blocks. You can select only **Address**, which indicates CIDR blocks.
ENI Type	Specify the type of data transfer that the security group rule controls. ○ **Internal**: controls data transfer within Alibaba Cloud. ○ **External**: controls data transfer over the Internet.
Authorization IP Addresses	Specify the CIDR blocks that you want the security group rule to accept or block. You can specify at most 10 CIDR blocks.
Enable Automatically Configure Routers	Specify whether to automatically propagate routes. The feature is enabled by default.
Description	Enter a description for the security group rule. The description must be 2 to 256 characters in length, and cannot start with `http://` or `https://`. You can leave this parameter empty.

24.1.7.2.4. Delete an SSL server

This topic describes how to delete an SSL server.

Procedure

1. Log on to the VPN Gateway console.
2.
3.
4. On the **SSL Servers** page, find the target SSL server, and then click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

24.1.7.3. Manage an SSL client certificate

24.1.7.3.1. Create an SSL client certificate

After you create an SSL server, you must create an SSL client certificate based on the configuration of the SSL server before you can establish an SSL-VPN connection.

Prerequisites

An SSL server is created. For more information, see Create an SSL server.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose **VPN > SSL Clients**.
3. Select the region where the SSL client is deployed.
4. On the **SSL Clients** page, click **Create Client Certificate**.
5. On the **Create SSL Client Certificate** page, configure the client certificate based on the following

information, and then click **Submit**.

Parameter	Description
Organization	Select the organization to which the SSL client belongs.
Resource Set	Select the resource set to which the SSL client belongs.
Region	Select the region where the SSL client is deployed.
Zone	Select the zone where the SSL client is deployed.
Name	Enter a name for the SSL client certificate. The name must be 2 to 128 characters in length, and can contain letters, digits, hyphens (-), and underscores (_). It must start with a letter or Chinese character and cannot start with `http://` or `https://`.
VPN Gateway	Select the VPN gateway that is used to establish the SSL-VPN connection.
SSL Server	Select the SSL server to be associated with the SSL client.

24.1.7.3.2. Download an SSL client certificate

This topic describes how to download an SSL client certificate. Before you use an SSL client to initiate an SSL-VPN connection, you must import the SSL client certificate to the SSL client.

Prerequisites

An SSL client certificate is created. For information, see Create an SSL client certificate.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose **VPN > SSL Clients**.
3. Select the region where the SSL client is deployed.
4. On the **SSL Clients** page, find the target SSL client certificate, and then click **Download** in the **Actions** column.

24.1.7.3.3. Delete an SSL client certificate

This topic describes how to delete an SSL client certificate.

Procedure

1. Log on to the VPN Gateway console.
2.
3.
4. On the **SSL Clients** page, find the target SSL client certificate, and then click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

24.1.8. Configure IPsec-VPN connections

24.1.8.1. Configuration overview

This topic describes how to connect a VPC to an on-premises data center through IPsec-VPN.

Prerequisites

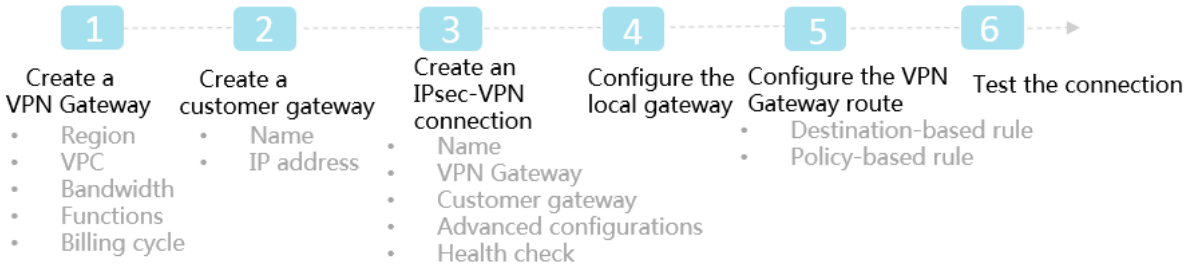
Before creating a site-to-site VPN connection, make sure the following conditions are met:

- The protocols IKEv1 and IKEv2 are supported by the gateway device of the on-premises data center.

  IPsec-VPN supports IKEv1 and IKEv2 protocols. Devices that support these two protocols can connect to Alibaba Cloud VPN Gateway, including devices of Huawei, H3C, Hillstone, SANGFOR, Cisco ASA, Juniper, SonicWall, Nokia, IBM, and Ixia.
- A static public IP address is configured for the local gateway.
- The IP address ranges of the VPC and on-premises data center to be connected do not conflict with each other.

Procedure

The following figure shows the procedure of connecting a VPC to an on-premises data center through IPsec-VPN.



1. Create a VPN Gateway

   Enable the IPsec-VPN function. Up to 10 IPsec-VPN connections can be established in a VPN Gateway.
2. Create a customer gateway

   By creating a customer gateway, you can register the local gateway to Alibaba Cloud and connect the customer gateway to the VPN Gateway. A customer gateway can be connected to multiple VPN Gateways.
3. Create an IPsec connection

   An IPsec connection is a VPN channel established between a VPN Gateway and a customer gateway. The encrypted communication between the VPN Gateway and the on-premises data center can be achieved only after the IPsec connection is established.
4. Configure the local gateway

   You need to load the VPN Gateway configurations to the local gateway device.
5. Configure the VPN Gateway route

   You need to configure a route in the VPN Gateway and publish it to the VPC route table.
6. Test the connection

   Log on to an ECS instance (without a public IP address) in the connected VPC. **ping** the private IP address of a server in the on-premises data center to check whether the connection is established.

24.1.8.2. Manage an IPsec-VPN connection

24.1.8.2.1. Create an IPsec-VPN connection

This topic describes how to create an IPsec-VPN connection. After you create a VPN gateway and a customer gateway, you can create an IPsec-VPN connection between the two gateways for encrypted data transmission.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose **VPN > IPsec Connections**.
3. Select the region where you want to create the IPsec-VPN connection.
4. On the **IPsec Connections** page, click **Create IPsec Connection**.
5. On the **Create IPsec Connection** page, configure the IPsec-VPN connection based on the following information and click **Submit**.

Parameter	Description
Organization	Select the organization to which the IPsec-VPN connection belongs.
Resource Set	Select the resource set to which the IPsec-VPN connection belongs.
Region	Select the region where the IPsec-VPN connection is established.
Zone	Select the zone where the IPsec-VPN connection is established.
Name	Enter a name for the IPsec-VPN connection. The name must be 2 to 128 characters in length, and can contain letters, digits, hyphens (-), and underscores (_). It must start with a letter or Chinese character and cannot start with `http://` or `https://`.
VPN Gateway	Select the VPN gateway to be connected through the IPsec-VPN connection.
Customer Gateway	Select the customer gateway to be connected through the IPsec-VPN connection.
Source CIDR Block	Enter the CIDR block of the VPC network to be connected to the on-premises data center. The CIDR block is used during phase 2 negotiation. You can add more than one CIDR blocks only if IKEv2 is used.
Destination CIDR Block	Enter the CIDR block of the on-premises data center to be connected to the VPC network. This CIDR block is used during phase 2 negotiation. You can add more than one CIDR blocks only if IKEv2 is used.
Immediate Effect	Specify whether to start connection negotiations immediately. ○ **Yes**: Negotiate immediately after the configuration is complete. ○ **No**: Negotiate when traffic is detected in the IPsec-VPN tunnel.
Advanced Settings	Specify whether to customize the advanced settings. ○ **Default**: Use default settings. ○ **Configure**: Use custom settings.
Advanced configuration: IKE configuration	
Pre-shared Key	Enter the pre-shared key used for authentication between the VPN gateway and customer gateway. You can specify a key, or use the default key that is randomly generated by the system.

Parameter	Description
Version	The version of the IKE protocol. Select an IKE version. Compared with IKEv1, IKEv2 simplifies the process of Security Association (SA) negotiation and provides better support for scenarios where an SSL-VPN connection is established with multiple subnets. We recommend that you select IKEv2.
Negotiation Mode	The negotiation mode of IKEv1. ○ Main: This mode offers higher security. ○ Aggressive: This mode is faster than the main mode. Negotiations are more likely to succeed in this mode. Connections negotiated in both modes ensure the same security level of data transmission.
Encryption Algorithm	Select the encryption algorithm used during phase 1 negotiation. Supported algorithms are aes, aes192, aes256, des, and 3des.
Authentication Algorithm	Select the authentication algorithm used during phase 1 negotiation. Supported algorithms are sha1, and md5.
DH Group	Select the Diffie-Hellman key exchange algorithm used during phase 1 negotiation.
SA Life Cycle (Seconds)	Specify the lifecycle of the SA after phase 1 negotiation succeeds. Default value: 86,400.
LocalId	The ID of the VPN gateway used during phase 1 negotiation. The default value is the public IP address of the VPN gateway. If you set LocalId to a FQDN, we recommend that you set Negotiation Mode to Aggressive.
RemoteId	The ID of the customer gateway used during phase 1 negotiation. The default value is the public IP address of the customer gateway. If you set RemoteId to a FQDN, we recommend that you select set Negotiation Mode to Aggressive.
Advanced configuration: IPSec configuration	
Encryption Algorithm	Select the encryption algorithm used during phase 2 negotiation. Supported algorithms are aes, aes192, aes256, des, and 3des.
Authentication Algorithm	Select the authentication algorithm used during phase 2 negotiation. Supported algorithms are sha1, and md5.
DH Group	Select the Diffie-Hellman key exchange algorithm for phase 2 negotiations. ○ If you select a Diffie-Hellman group, the perfect forward secrecy (PFS) feature is enabled and each negotiation requires a new key. Therefore, you must also enable PFS for the client. ○ For clients that do not support PFS, select disabled.
SA Life Cycle (Seconds)	Specify the lifecycle of the SA after phase 2 negotiation succeeds. Default value: 86,400.

24.1.8.2.2. Modify an IPsec-VPN connection

This topic describes how to modify the name, advanced settings, and health check for an IPsec-VPN connection.

Prerequisites

An IPsec-VPN connection is created. For more information, see Create an IPsec-VPN connection.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose **VPN > IPsec Connections**.
3. In the top navigation bar, select the region where the IPsec-VPN connection is created.
4. On the **IPsec Connections** page, find the target IPsec-VPN connection and then click **Edit** in the **Actions** column.
5. In the **Modify IPsec Connections** dialog box, modify the name, advanced settings, and health check, and click **Submit**.

24.1.8.2.3. Download the configuration file of an IPsec-VPN connection

This topic describes how to download the configurations of an IPsec-VPN connection, and load the configurations to the customer gateway device after an IPsec-VPN connection is configured.

Prerequisites

An IPsec-VPN connection is created. For more information, see Create an IPsec-VPN connection.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose **VPN > IPsec Connections**.
3. In the top navigation bar, select the region where the IPsec-VPN connection is created.
4. On the **IPsec Connections** page, find the target IPsec-VPN connection, and then choose **More > Download Configuration** in the **Actions** column.

> **Note** RemoteSubnet and LocalSubnet in the downloaded configurations are opposite to RemoteSubnet and LocalSubnet that you specify when you create an IPsec-VPN connection. For a VPN gateway, RemotSubnet refers to the CIDR block of the on-premises data center and LocalSubnet refers to the CIDR block of the VPC network. For a customer gateway, LocalSubnet refers to the CIDR block of the on-premises data center and RemoteSubnet refers to the CIDR block of the VPC network.

24.1.8.2.4. Configure a routing group

This topic describes how to configure routing groups to control the inbound and outbound traffic of ECS instances in a routing group after an IPsec-VPN connection is created.

Procedure

1. Log on to the VPN Gateway console.
2.
3.
4. On the **IPsec Connections** page, find the target IPsec-VPN connection, and then choose **More > Configure Routing Group** in the **Actions** column.
5. In the **Configure Routing Group** dialog box, set the following parameters, and then click **Submit**.

Parameter	Description
Routing Group	Select the routing group to which you want to add the routing group rule.

Parameter	Description
Rule Direction	Select the direction in which the rule is applied. ◦ **Outbound**: from the ECS instances in the current routing group to other ECS instances on Alibaba Cloud or resources on the Internet. ◦ **Inbound**: from other ECS instances on Alibaba Cloud or resources on the Internet to the ECS instances in the current routing group.
Authorization Policy	Select an authorization policy. ◦ **Allow**: accept requests received on the specified ports. ◦ **Deny**: discard requests received on the specified ports without returning messages. If you specify different authorization policies for two routing group rules but the other settings are the same, the **Deny** rule prevails over the **Allow** rule.
Protocol Type	Select a protocol type.
Port Range	Select a port range for the routing group rule. The port range depends on the protocol type: ◦ When you set Protocol to **All**, this parameter displays **-1/-1**, which indicates all ports. You cannot specify a port range if you select this protocol type. ◦ When you set Protocol to **TCP**, you can specify a port range in the <start port number>/<end port number> format. Valid port numbers: 1 to 65535. To specify a single port, set the start and end port numbers to the same value. For example, use 22/22 to indicate port 22. ◦ When Protocol is set to **UDP**, specify a port range in the <start port number>/<end port number> format. Valid port numbers: 1 to 65535. To specify a single port, set the start and end port numbers to the same value. For example, use 3389/3389 to indicate port 3389. ◦ When Protocol is set to **ICMP**, this parameter displays **-1/-1**, which indicates all ports. You cannot set a port range if you select this protocol type. ◦ When Protocol is set to **GRE**, this parameter displays **-1/-1**, which indicates all ports. You cannot set a port range if you select this protocol type.
Priority	Set the priority of the rule. Valid values: 1 to 100. The default value is 1, which indicates the highest priority.
Authorization Type	Select the authorization type of the routing group rule. You can select only **Address**.
NIC type	Select a NIC type. ◦ **Internal**: Control inbound and outbound traffic within Alibaba Cloud. ◦ **External**: Control inbound and outbound traffic over the Internet.
Authorized IP Addresses	Select the CIDR blocks to be authorized. You can specify up to 10 CIDR blocks at a time.
Automatically Configure Routers	Specify whether to automatically configure routers.

Parameter	Description
Description	The description of the routing group rule. The description must be 2 to 256 characters in length, and cannot start with `http://` or `https://`. You can left this parameter empty.

24.1.8.2.5. View IPsec-VPN connection logs

This topic describes how to view IPsec-VPN connection logs that are generated within the last 30 days to troubleshoot connection errors. You can query log data generated within 10 minutes.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose **VPN > IPsec Connections**.
3. In the top navigation bar, select the region where the IPsec-VPN connection is created.
4. On the **IPsec Connections** page, find the target IPsec-VPN connection, and then choose **More > View Logs** in the **Actions** column.
5. In the **IPsec Connection Logs** dialog box, set the time range and query the logs.

24.1.8.2.6. Delete an IPsec-VPN connection

This topic describes how to delete an IPsec-VPN connection.

Procedure

1. Log on to the VPN Gateway console.
2.
3.
4. On the **IPsec Connections** page, find the target IPsec-VPN connection, and then click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

24.1.8.3. MTU notes

The maximum transmission unit (MTU) is the size (in bytes) of the largest packet supported by the network layer protocol (such as TCP), with headers and data included.

Network packets sent over IPsec tunnels are encrypted and then encapsulated in external packets for routing. Because an encapsulated internal packet itself must fit the MTU of the corresponding external packet, the MTU of the internal packet must be smaller.

Gateway MTU and system MTU

You must configure the MTU limit of the local VPN Gateway to not more than 1,400 bytes. We recommend that you set the MTU to 1,400 bytes.

For TCP traffic, the maximum length of data that can be carried by each packet segment can be negotiated by the sender and receiver when they communicate based on the maximum segment size (MSS).

25.Elastic IP Address

25.1. User Guide

25.1.1. What is an EIP?

This topic provides an overview of Elastic IP Address. An elastic IP address (EIP) is a public IP address that you can purchase and hold as an independent resource. You can associate an EIP with an Elastic Compute Service (ECS) instance deployed in a virtual private cloud (VPC), an internal Server Load Balancer (SLB) instance deployed in a VPC, or a secondary elastic network interface (ENI) attached to a VPC. You can also associate an EIP with a NAT gateway, or a High-Availability Virtual IP Address (HAVIP).

An EIP is also a NAT IP address that is provisioned in a public-facing gateway of Alibaba Cloud and is mapped to the associated cloud resource with NAT. After an EIP is associated with a cloud resource, the cloud resource can connect to the Internet by using this EIP.

Line types

EIPs support BGP (Multi-ISP) lines and BGP (Multi-ISP) Pro lines. The following table describes the differences between BGP (Multi-ISP) and BGP (Multi-ISP) Pro.

Item	BGP (Multi-ISP)	BGP (Multi-ISP) Pro
Advantages	BGP (Multi-ISP) provides high-quality and high-bandwidth BGP lines. • Up to 89 high-quality BGP lines are available worldwide. • Direct connections can be established in all regions of the Chinese mainland through lines of the following Internet Service Providers (ISPs): China Telecom, China Unicom, China Mobile, China Mobile Tietong, China Netcom, China Education and Research Network (CERNET), National Radio and Television Administration, Dr. Peng Telecom & Media Group, and Founder Broadband Network. • You can use BGP (Multi-ISP) with EIP bandwidth plans to obtain bandwidth of 100 Gbit/s and higher.	BGP (Multi-ISP) Pro lines optimize data transmission to the Chinese mainland and improve connection quality for international services. Compared with BGP (Multi-ISP), when BGP (Multi-ISP) Pro lines provide services to users in the Chinese mainland (excluding data centers), cross-border connections are established by using Chinese mainland ISP services. This reduces network latency.
Region	All regions	China (Hong Kong)

Benefits

EIPs have the following benefits:

- Independent purchase and possession

  You can purchase and hold an EIP as an independent resource. EIPs are not bundled with other computing or storage resources.

- Flexible association

  You can dissociate an EIP from a cloud resource and then release the EIP if the EIP is no longer needed.

- Configurable network capabilities

  You can adjust the peak bandwidth of an EIP at any time. The bandwidth changes take effect immediately.

25.1.2. Log on to the EIP console

This topic describes how to log on to the Apsara Uni-manager Management Console to manage your elastic IP addresses (EIPs). The Google Chrome browser is used as an example.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Login**.
4. In the top menu bar, choose **Products > Networking > Elastic IP Address**.

25.1.3. Quick start

25.1.3.1. Tutorial overview

This topic provides an overview of the tutorial that guides you in creating an EIP and associating an EIP with an ECS instance to allow the ECS instance to access the Internet.

This tutorial walks you through the following tasks:

1. Apply for new EIPs

   An EIP is a public IP address that you can purchase and hold as an independent resource. To get started, you must create an EIP.

2. Associate an EIP with an ECS instance

   You can associate an EIP with an ECS instance deployed in a VPC to enable the ECS instance to connect to the Internet.

3. Disassociate an EIP from a cloud resource

   You can disassociate an ECS instance from an EIP when the ECS instance no longer requires access to the Internet.

4. Release an EIP

   You can release an EIP if it is no longer needed.

25.1.3.2. Apply for EIPs

This topic describes how to apply for elastic IP addresses (EIPs). An EIP is a public IP address that you can purchase and hold as an independent resource.

Procedure

1. Log on to the EIP console.
2. On the **Elastic IP Addresses** page, click **Create EIP**.
3. On the **Create Elastic IP Address** page, set the following parameters and click **Submit**.

Parameter	Description
Organization	Select the organization to which the EIP belongs.
Resource Set	Select the resource group to which the EIP belongs.
Region	Select a region for the EIP. Make sure that the EIP and the cloud resources to be associated with the EIP are deployed in the same region.
Zone	Select a zone for the EIP.
Internet Connection Type	Select a line type for the EIP.
Network Type	Select a network type for the EIP. ◦ **Public Network**: The EIP is used to enable communication on the Internet. ◦ **Hybrid Cloud**: The EIP is used to enable communication within a hybrid cloud. For example, you must select an EIP of this type to allow your server in the on-premises data center to access the Internet by using Source Network Address Translation (SNAT) and Destination Network Address Translation (DNAT).
Service IP	Enter the IP address that you want to request. Make sure that the specified IP address is an IPv4 address and the IP address is not used by another account. Otherwise, the EIP cannot be allocated. **Note** If you do not specify an EIP, the system automatically assigns one.
Max Bandwidth	Specify the maximum bandwidth of the EIP. Unit: Mbit/s.

25.1.3.3. Associate an EIP with an ECS instance

This topic describes how to associate an EIP with an ECS instance deployed in a VPC. ECS instances that are associated with EIPs can communicate with the Internet.

Prerequisites

You have created an ECS instance. For more information, see the **Create an instance** topic of **Quick start** in the *Apsara Stack Elastic Compute Service User Guide*.

Procedure

1. Log on to the EIP console.

2. In the top navigation bar, select the region where the EIP is deployed.
3. On the **Elastic IP Addresses** page, find the target EIP, and then click **Bind** in the **Actions** column.
4. In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

Parameter	Description
Instance Type	Select **ECS Instance**.
ECS Instance	Select the ECS instance to be associated with the EIP. When you select an ECS instance, note the following points: ○ The ECS instance must be deployed in a VPC. ○ The ECS instance must be running or stopped. ○ An ECS instance can be associated with only one EIP. ○ The ECS instance and the EIP must reside in the same region. ○ The ECS instance is not associated with a public IP address or another EIP.

25.1.3.4. Disassociate an EIP from a cloud resource

This topic describes how to dissociate an EIP from a cloud resource when this cloud resource no longer needs to communicate with the Internet.

Procedure

1. Log on to the EIP console.
2.
3. On the **Elastic IP Addresses** page, find the target EIP, and click **Unbind** in the **Actions** column.
4. In the **Unbind Elastic IP Address** dialog box, click **OK**.

25.1.3.5. Release an EIP

This topic describes how to release an Elastic IP address (EIP).

Prerequisites

The EIP is disassociated from all instances. For more information, see Disassociate an EIP from a cloud resource.

Procedure

1. Log on to the EIP console.
2.
3. On the **Elastic IP Addresses** page, find the target EIP, move the mouse pointer over **More operations** in the **Actions** column, and then click **Release**.
4. In the **Release an EIP** dialog box, click **OK**.

25.1.4. Manage EIPs

25.1.4.1. Create a EIP

This topic describes how to create a EIP. An EIP is a public IP address that you can purchase and hold as an independent resource.

Procedure

1. Log on to the EIP console.
2. On the **Elastic IP Addresses** page, click **Create EIP**.
3. On the **Create Elastic IP Address** page, set the parameters and click **Submit**. The following table describes the configuration parameters.

Parameter	Description
Organization	Select the organization to which the EIP belongs.
Resource Set	Select the resource set to which the EIP belongs.
Region	Select a region for the EIP. Make sure that the EIP and the cloud resource to be associated with the EIP are located in the same region.
Zone	Select the zone to which the EIP belongs.
Line Type	Select the line type of the EIP.
Network Type	Select the network type of communication for which the EIP will be used. ○ **Public Network**: Use the EIP to enable communication with the Internet. ○ **Hybrid Cloud**: Use the EIP to enable communication within a hybrid cloud. For example, if you need to establish network connections from an on-premises data center to the Internet by using source network address translation (SNAT) and destination network address translation (DNAT), you must select this type for your EIP.
Service IP	Specify a specific EIP. Make sure that the specified IP address is an IPv4 address and is not taken by another account. Otherwise, the EIP cannot be allocated. **Note** If you do not specify a service IP, a random EIP will be assigned to you by the system.
Peak Bandwidth	Specify the maximum bandwidth for the EIP. Unit: Mbit/s.

25.1.4.2. Bind an EIP to a cloud instance

25.1.4.2.1. Associate an EIP with an ECS instance

This topic describes how to associate an EIP with an ECS instance deployed in a VPC. ECS instances that are associated with EIPs can communicate with the Internet.

Prerequisites

You have created an ECS instance. For more information, see the **Create an instance** topic of **Quick start** in the *Apsara Stack Elastic Compute Service User Guide*.

Procedure

1. Log on to the EIP console.
2. In the top navigation bar, select the region where the EIP is deployed.

3. On the **Elastic IP Addresses** page, find the target EIP, and then click **Bind** in the **Actions** column.
4. In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

Parameter	Description
Instance Type	Select **ECS Instance**.
ECS Instance	Select the ECS instance to be associated with the EIP. When you select an ECS instance, note the following points: ○ The ECS instance must be deployed in a VPC. ○ The ECS instance must be running or stopped. ○ An ECS instance can be associated with only one EIP. ○ The ECS instance and the EIP must reside in the same region. ○ The ECS instance is not associated with a public IP address or another EIP.

25.1.4.2.2. Associate an EIP with an SLB instance

This topic describes how to associate an EIP with an SLB instance. After the association, the SLB instance can distribute requests from the Internet.

Prerequisites

You have created an SLB instance. For more information, see the **Create an SLB instance** topic of **Quick start** in the *Apsara Stack Server Load Balancer User Guide*.

Procedure

1.
2. Log on to the EIP console.
3.
4.
5. In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

Parameter	Description
Instance Type	Select **SLB Instance**.
SLB Instance	Select the SLB instance to be associated with the EIP. When you select an SLB instance, note the following points: ○ The SLB instance must be deployed in a VPC. ○ The SLB instance and the EIP must reside in the same region. ○ An SLB instance can be associated with only one EIP.

25.1.4.2.3. Associate an EIP with an HAVIP

This topic describes how to associate an EIP with an HAVIP. After the association, the HAVIP can be used to connect to the Internet.

Prerequisites

You have created an HAVIP. For more information, see the **Create an HAVIP** topic of **HAVIPs** in the *Apsara Stack*

Virtual Private Cloud User Guide.

Procedure

1. Log on to the EIP console.
2.
3.
4. In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

Parameter	Description
Instance Type	Select **HaVip Address**.
HaVip Address	Select the HAVIP to be associated with the EIP. When you select an HAVIP, note the following points: ○ The HAVIP and the EIP must reside in the same region. ○ The HAVIP must be available or allocated. ○ An HAVIP can be associated with only one EIP.

25.1.4.2.4. Associate an EIP with a NAT gateway

This topic describes how to associate an EIP with a NAT gateway. After the association, you can use the EIP to configure DNAT and SNAT entries.

Prerequisites

You have created a NAT gateway. For more information, see the **Create a NAT gateway** topic of **Quick start** in the *Apsara Stack NAT Gateway User Guide*.

Procedure

1. Log on to the EIP console.
2.
3.
4. In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

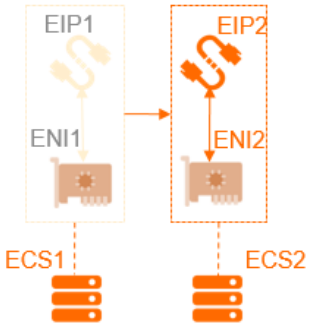
Parameter	Description
Instance Type	Select **NAT Gateway**.
NAT Gateway	Select the NAT gateway to be associated with the EIP. When you select a NAT gateway, note the following points: ○ The NAT gateway and the EIP must reside in the same region. ○ A NAT gateway can be associated with a maximum of 20 EIPs.

25.1.4.2.5. Bind an EIP to a secondary ENI

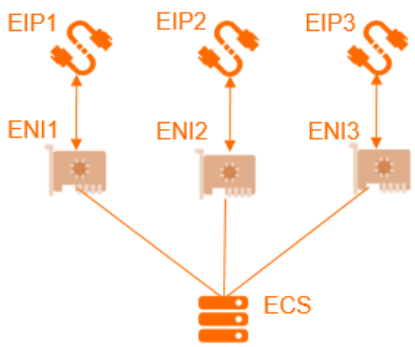
25.1.4.2.5.1. Overview

This topic provides an overview of EIP associations with ENIs. You can associate EIPs with ENIs to build a highly robust, flexible, and scalable IT solution. This also allows a cloud server to use multiple public IP addresses.

An ENI is assigned with a private IP address. After you associate an EIP with an ENI, the ENI has both a private IP address and a public IP address. When you migrate an ENI that is associated with an EIP from an ECS instance to another ECS instance, the public and private IP addresses are also migrated. This provides a highly reliable and available IP migration solution for cloud servers that use both public and private IP addresses.



To enable an ECS instance to use multiple public IP addresses, you can associate the ECS instance with multiple ENIs, and then associate each ENI with an EIP. You can use these public IP addresses along with security group rules to provide external services.



Association modes

You can associate an EIP with an ENI in one of the following two modes:

- NAT mode
- Cut-through mode

The following table lists the distinguishing features of each mode.

Feature	NAT mode	Cut-through mode
Support for displaying the associated EIPs in the ENI information	No	Yes **Note** You can run the **ifconfig** or **ipconfig** command to query the EIP associated with the corresponding ENI.
Supported ENIs that can be associated with EIPs	Primary and secondary ENIs	Secondary ENIs
The maximum number of EIPs that can be associated with a primary ENI	1	EIP associations with primary ENIs are not supported.

Feature	NAT mode	Cut-through mode
The maximum number of EIPs that can be associated with a secondary ENI	Depends on the number of private IP addresses of the secondary ENI. **Note** A one-to-one relationship is implemented between EIPs and the private IP addresses of a secondary ENI. For example, if a secondary ENI has 10 private IP addresses, a maximum of 10 EIPs can be associated with this ENI.	1 **Note** In the cut-through mode, an EIP can only be associated with the primary private IP address of a secondary ENI.
Support for the secondary ENI to deliver networking connectivity within the private network after the ENI is associated with an EIP	Yes	No
Supported protocols	EIPs that are associated in the NAT mode do not support the protocols that require connection management from NAT application-level gateway (ALG), such as H.323, Session Initiation Protocol (SIP), Domain Network System (DNS), and Real Time Streaming Protocol (RTSP).	All IP protocols, such as H.323, SIP, DNS, RTSP, File Transfer Protocol (FTP), and Trivial File Transfer Protocol (TFTP).

25.1.4.2.5.2. Associate an EIP with an ENI in the NAT mode

This topic describes how to associate an EIP with an ENI in the NAT mode. With this association, both the public and private IP addresses of the ENI can be used to provide networking connectivity. EIPs associated in the NAT mode cannot be displayed in the ENI information.

Prerequisites

Before you associate an EIP with an ENI in the NAT mode, make sure that the following conditions are met:

- You have created a secondary ENI that is attached to a VPC. The secondary ENI and the EIP to be associated reside in the same region. For more information, see the **Create an ENI** topic of **Elastic Network Interfaces** in the *Apsara Stack Elastic Compute Service User Guide*.
- The secondary ENI is not associated with an ECS instance.

  If the secondary ENI is associated with an ECS instance, you must disassociate it from the ECS instance. You can re-associate the ENI with the ECS instance after you have associated the ENI with the intended EIP in the NAT mode. For more information, see the **Unbind a secondary ENI from an instance** topic of **Elastic Network Interfaces** in the *Apsara Stack Elastic Compute Service User Guide*.

Context

The NAT mode has the following characteristics:

- The number of EIPs with which a secondary ENI can be associated depends on the number of private IP addresses of this secondary ENI.
- When an EIP is associated with an ENI in the NAT mode, both the private and public IP addresses of this ENI are available.
- The EIP associated with an ENI cannot be viewed in the operating system. To query the EIP associated with an ENI, call the DescribeEipAddresses operation.

- EIPs that are associated in the NAT mode do not support the protocols that require connection management from NAT ALG, such as H.323, SIP, DNS, RTSP, and TFTP.

Procedure

1. Log on to the EIP console.
2. In the top navigation bar, select the region of the EIP.
3. On the **Elastic IP Addresses** page, find the target EIP and click **Bind** in the **Actions** column.
4. In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

Parameter	Description
IP Address	Displays the target EIP.
Instance Type	Select **Secondary ENI**.
Mode	Select **NAT Mode**.
Secondary ENI	Select the secondary ENI to be associated with the EIP.

25.1.4.2.5.3. Associate an EIP with an ENI in the cut-through mode

This topic describes how to associate an EIP with an ENI in the cut-through mode. After the association, the EIP replaces the private IP address of the secondary ENI, which changes the secondary ENI to an Internet interface. You can view the EIP in the ENI information.

Prerequisites

Before you associate an EIP with an ENI in the cut-through mode, make sure that the following conditions are met:

- The secondary ENI is attached to a VPC.
- The secondary ENI and the EIP reside in the same region.
- The secondary ENI is not associated with an ECS instance.

  If the secondary ENI is associated with an ECS instance, you must disassociate it from the ECS instance. You can re-associate the ENI with the ECS instance after you have associated the ENI with the intended EIP in the cut-through mode. For more information, see the **Unbind a secondary ENI from an instance** topic of **Elastic Network Interfaces** in the *Apsara Stack Elastic Compute Service User Guide*.
- A secondary ENI can be associated with only one EIP.

Context

An EIP is essentially a NAT IP address. After you associate an EIP with an ENI in the NAT mode, the public IP address of the ENI is provisioned in a gateway rather than the ENI of the associated ECS instance. Therefore, the public IP address cannot be displayed in the operating system of the ECS instance, which only shows the private IP address in the ENI information. This complicates operations and maintenance because you must keep track of the mappings between network interfaces or servers and public IP addresses. In addition, EIPs that are associated with ENIs in the NAT mode do not support protocols such as H.323, SIP, DNS, or RTSP.

To solve the preceding problems, you can associate an EIP with an ENI in **cut-through mode**. In the cut-through mode:

- The EIP replaces the private IP address of the secondary ENI. This changes the secondary ENI to an Internet interface which can no longer deliver networking connectivity within the private network.

- You can view the EIP in the ENI information. To query the EIP associated with the corresponding ENI, you can call the **ifconfig** or **ipconfig** operation.
- EIPs that are associated in the cut-through mode support all Internet protocols, such as H.323, SIP, DNS, RTSP, FTP, and TFTP.

Procedure

1. Log on to the EIP console.
2. Select the region of the EIP.
3.
4. In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

Parameter	Description
Instance Type	Select **Secondary ENI**.
Binding mode	Select **Cut-Through Mode**.
Secondary ENI	Select the secondary ENI to be associated with the EIP.

> **Notice** Make sure that the selected secondary ENI is not associated with an ECS instance.

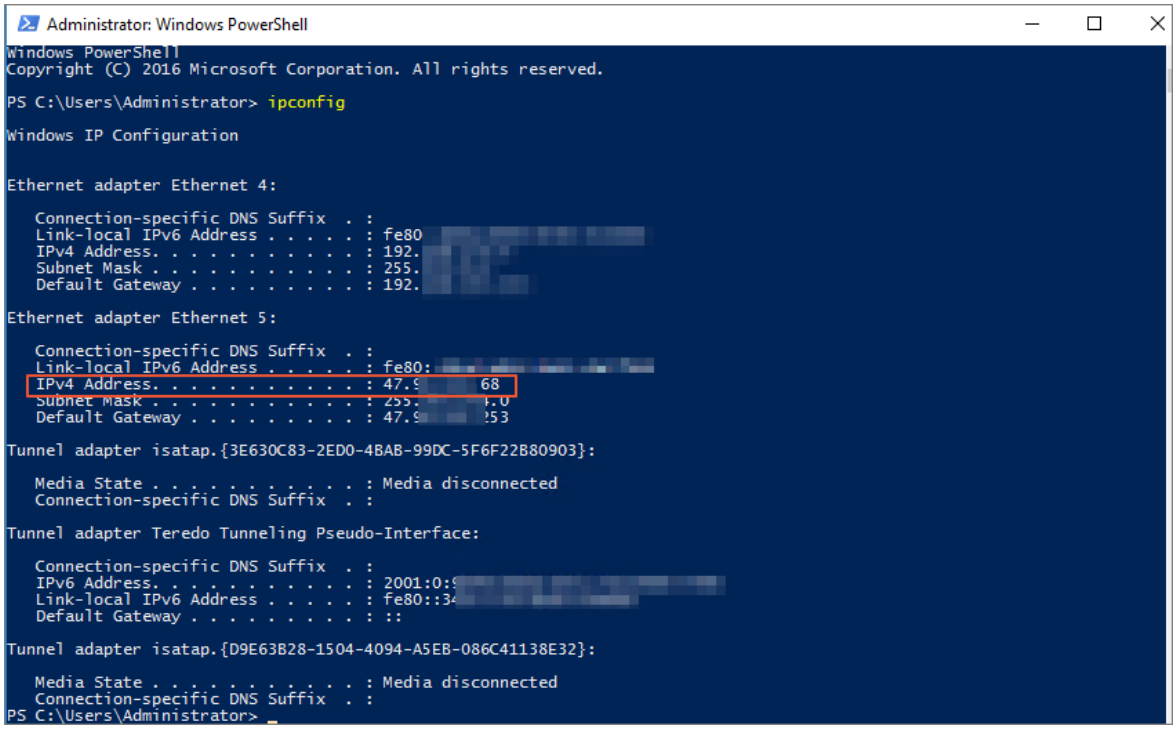
5. Return to the Elastic IP Addresses page and click the ID of the associated ENI.
6. Find the target ENI from the list of ENIs. In the **Actions** column, click **Bind** to associate the ENI with an ECS instance.

> **Note**
> - The maximum number of ENIs that can be associated with an ECS instance depends on the specification of the ECS instance. For more information, see the **Instance types** topic of **What is ECS** in the *Apsara Stack Elastic Compute Service User Guide*.
> - After you associate a secondary ENI with an ECS instance, you must enable Dynamic Host Configuration Protocol (DHCP) for the secondary ENI and then restart the ENI. Otherwise, the configuration made for the cut-through mode will not take effect.
> - After the configuration is complete, a route entry is automatically created for the ECS instance with the secondary ENI as the egress interface. The priority of this route entry is lower than that of the route with the primary ENI as the egress interface. Therefore, you can adjust the priority of these entries based on your service requirements.

7. Log on to the ECS instance by using the associated EIP to check the network configuration of the ECS instance.

> **Note** Make sure that the security group rules of the ECS instance allow remote access.

As shown in the following figure, the IPv4 address of the ECS instance is changed to the associated EIP.



25.1.4.3. Upgrade a subscription EIP

This topic describes how to upgrade the bandwidth of EIP. Bandwidth upgrades take effect immediately.

Procedure

1. Log on to the EIP console.
2.
3. On the **Elastic IP Addresses** page, find the target EIP and choose **More > Upgrade** in the **Actions** column.
4. On the **Change Specifications** page, set a new peak bandwidth for the EIP, and then click **Submit**.

25.1.4.4. Disassociate an EIP from a cloud resource

This topic describes how to dissociate an EIP from a cloud resource when this cloud resource no longer needs to communicate with the Internet.

Procedure

1. Log on to the EIP console.
2.
3. On the **Elastic IP Addresses** page, find the target EIP, and click **Unbind** in the **Actions** column.
4. In the **Unbind Elastic IP Address** dialog box, click **OK**.

25.1.4.5. Release an EIP

This topic describes how to release an Elastic IP address (EIP).

Prerequisites

The EIP is disassociated from all instances. For more information, see Disassociate an EIP from a cloud resource.

Procedure

1. Log on to the EIP console.
2.
3. On the **Elastic IP Addresses** page, find the target EIP, move the mouse pointer over **More operations** in the **Actions** column, and then click **Release**.
4. In the **Release an EIP** dialog box, click **OK**.

26.Express Connect

26.1. User Guide

26.1.1. What is Express Connect?

This topic provides an overview of Express Connect. Express Connect allows you to establish private connections to enable fast, stable, and secure communication between different networking environments. You can use Express Connect to ensure network stability and prevent data breaches.

Features

You can use a leased line provided by an Internet Service Provider (ISP) to establish a physical connection between your data center and an Alibaba Cloud access point. After the physical connection is established, you can create a virtual border router (VBR) to connect your data center with Alibaba Cloud to build a hybrid cloud.

The physical connections of Express Connect do not traverse the Internet, and therefore feature faster speeds, lower latency, greater security, and higher reliability compared with Internet connections.

Express Connect enables you to create a peering connection between two Virtual Private Clouds (VPCs) as a channel for private communication.

Benefits

Express Connect provides the following benefits:

- High-speed interconnections

  Powered by the network virtualization technology of Alibaba Cloud, Express Connect allows networks to connect and exchange traffic at high speeds within internal networks without carrying traffic across the Internet. The impact of distance on network performance is minimized to ensure low-latency and high-bandwidth communication.

- Stability and reliability

  Built on the state-of-the-art infrastructure of Alibaba Cloud, Express Connect guarantees stable and reliable communication between networks.

- Security

  Express Connect implements cross-network communication at the network virtualization layer, where data is transmitted over separate and private channels within the infrastructure of Alibaba Cloud, mitigating the risks of data breaches.

- Buy-as-you-need service

  Express Connect delivers connectivity with a wide range of bandwidth options. You can choose based on the needs of your business to get the best value for your purchase.

26.1.2. Log on to the Express Connect console

This topic describes how to log on to the Apsara Uni-manager Management Console to manage your Express Connect services. The Google Chrome browser is used as an example.

Prerequisites

- Before you log on to the Apsara Uni-manager Management Console, you must obtain the URL of the console from the engineer that deploys the service.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar of the browser, enter the URL of the Apsara Uni-manager Management Console and press the Enter key.
2. Enter your username and password.

   Obtain the username and password that are used to log on to the console from the operations administrator.

   > **Note** If this is the first time you log on to the Apsara Uni-manager Management Console, you must change your password as prompted. For higher security, the password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters.
   > - Digits.
   > - Special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs ($), and percent signs (%).

3. Click **Log On**.
4. In the top navigation bar, choose **Products > Networking > Express Connect**.

26.1.3. VPC peering connections

26.1.3.1. What is a peering connection

You can create a peering connection to connect two virtual private clouds (VPCs). In addition, you can create a peering connection to connect a VPC and a virtual border router (VBR).

Initiator and acceptor

When you establish a peering connection between two VPCs or between a VPC and a VBR, one end of the peering connection functions as the initiator. The other end of the peering connection functions as the acceptor. Only the initiator can initiate the peering connection. The acceptor must wait for the initiator to initiate the peering connection. The concepts of initiator and acceptor are only used to control how a peering connection is established. Data transmission between the initiator and acceptor is bidirectional. Therefore, after the peering connection is established, both the initiator and acceptor can send and receive data.

When you connect two VPCs under the same account, you can specify the initiator and acceptor in the Express Connect console. After you specify the initiator and acceptor, the system automatically initiates a connection request and then establishes a peering connection. You do not need to manually send a connection request. To connect two VPCs under different accounts, you must manually send a request to establish a peering connection.

The following table describes the differences between the initiator and acceptor.

Item	Initiator	Acceptor
Whether the configuration of the peer is required before a peering connection is initiated	Yes	Yes
Initiate connection requests	Yes	No
Send messages to the peer after the peering connection is established	Yes	Yes

Connection stages and states

The initiator sends a connection request to the acceptor to establish a peering connection. After the acceptor accepts the request, the peering connection is established.

The following table describes the states of a peering connection at different stages.

> **Note** If you specify the initiator and acceptor when you create a peering connection, the system automatically sends a connection request and establishes the peering connection. The peering connection is activated on both the initiator and acceptor after the peering connection is established.

Stage	State of the peering connection on the initiator	State of the peering connection on the acceptor
Send a connection request from the initiator	Connecting	Accepting
Successfully establish a peering connection	Active	Active
Deactivate a peering connection	Deactivating	Deactivating
Close a peering connection	Inactive	Inactive
Resend a connection request	Activating	Activating

26.1.3.2. Connect two VPCs

This topic describes how to create a peering connection to connect two virtual private clouds (VPCs).

Procedure

1. Log on to the Express Connect console.
2. In the left-side navigation pane, choose **VPC Peering Connections > VPC-to-VPC**.
3. On the **VPC-to-VPC** page, click **Create Peering Connection**.
4. Set the following parameters to configure the peering connection.

Parameter	Description
Scenarios	Select **VPC-to-VPC** to create a peering connection between two VPCs.
Local Configurations	
Organization	Select the organization to which the source VPC belongs.
Resource Set	Select the resource group to which the source VPC belongs.
Region	Select the region where the source VPC is created.
Router Type	Use the default setting **vRouter** for this parameter.
Local VPC ID	Select the ID of the source VPC.
Peer Configurations	
Organization	Select the organization to which the destination VPC belongs.
Resource Set	Select the resource group to which the destination VPC belongs.
Peer Region	Select the region where the destination VPC is created.

Parameter	Description
Peer Router Type	Use the default setting **vRouter** for this parameter.
Peer VPC ID	Select the ID of the destination VPC.
Basic Settings	
Bandwidth	Specify the bandwidth of the peering connection.

5. Click **Submit**.

26.1.3.3. Connect a VBR to a VPC

This topic describes how to create a peering connection between a virtual border router (VBR) and a virtual private cloud (VPC).

Procedure

1. Log on to the Express Connect console.
2. In the left-side navigation pane, choose **VPC Peering Connections > VBR-to-VPC**.
3. On the **VBR-to-VPC** page, click **Create Peering Connection**.
4. On the **Create Peering Connection** page, set the following parameters to configure the peering connection.

Parameter	Description
Scenarios	Use the default setting **VBR-to-VPC** for this parameter.
Common Settings	
Organization	Select the organization to which the VBR belongs.
Resource Sets	Select the resource group to which the VBR belongs.
VBR-side Configurations	
Router Type	Use the default setting **VBR** for this parameter.
Region	Select the region where the VBR is created.
Endpoint	Select an access point.
VBR ID	Select the ID of the VBR.
Router Interface Specifications	Select the interface specification of the router. In this example, 1,000 Mbit/s is selected.
Router Interface Name	Enter a name for the router interface.
Description	Enter a description for the router.
VPC-side Configurations	
Router Type	Use the default setting **vRouter** for this parameter.

Parameter	Description
Region	Select the region where the VPC is created.
VPC ID	Select the ID of the VPC.
Router Interface Name	Enter a name for the router interface.
Health Check Source IP Address	Enter the source IP address of health checks. **Note** The source IP address of health checks must be an idle IP address that belongs to the CIDR block of a vSwitch in the VPC.
Health Check Destination IP Address	Enter the destination IP address of health checks. **Note** The destination IP address is the IP address of the customer-premises device that is connected to the VBR to which the physical connection is established.
Description	Enter a description for the VPC.

5. Click **Submit**.

26.1.3.4. Delete a peering connection

This topic describes how to delete a peering connection.

Procedure

1. Log on to the Express Connect console.
2. In the left-side navigation pane, choose **VPC Peering Connections > VPC-to-VPC**.
3. Find the target peering connection and choose > **Suspend Initiator** in the **Actions** column.
4. Find the target peering connection and choose > **Suspend Acceptor** in the **Actions** column.
5. Find the target peering connection and choose > **Delete** in the **Actions** column.
6. In the **Delete Peering Connection** dialog box, click **Confirm**.

26.1.4. Physical connections

26.1.4.1. What is a physical connection?

Express Connect provides a secure and convenient method to connect a data center to Alibaba Cloud. You can use a leased line provided by an Internet service provider (ISP) to establish a physical connection between your data center and an Alibaba Cloud access point. Physical connections do not pass through the Internet. Compared with Internet connections, physical connections are safer, faster, more reliable, and have lower latency.

You can use a point-to-point Ethernet connection or MPLS VPN connection. Only 10 Gbit/s single-mode fiber ports can be used to establish physical connections.

Connection methods

You can apply for a physical connection port to establish a physical connection.

Then, you can use the physical connection port to connect a data center to an Alibaba Cloud access point. The physical connection port is dedicated to the physical connection. You can apply for a dedicated physical connection port in the Express Connect console. For more information, see Create a physical connection.

26.1.4.2. Create a physical connection

This topic describes how to create a physical connection. To establish a physical connection, you must first confirm the access point of the data center. Then, apply for and activate a physical connection port and create a virtual border router (VBR) in the Express Connect console.

Step 1: Confirm the access point of the data center

An access point indicates where network services are provided. You can use multiple types of connection to connect a data center to Alibaba Cloud. When you select an access point, take note of the elements such as the region, service provider, and port type. When you select an access point, you must confirm the region of Apsara Stack that you want to access.

Step 2: Apply for a physical connection port in the console

To apply for a physical connection port, perform the following operations:

1. Log on to the Express Connect console.
2. In the left-side navigation pane, click **Exclusive Physical Connection**.
3. On the **Exclusive Physical Connection** page, click **Apply for New Interface**.
4. Set the following parameters and click **Submit** to configure the physical connection port.

Parameter	Description
Region	
Organization	Select the organization to which the physical connection belongs.
Resource Set	Select the resource group to which the physical connection belongs.
Region	Select the region where you want to establish the physical connection.
Basic Settings	
Physical Connection Name	Enter a name for the physical connection. The name must be 2 to 128 characters in length and can contain digits, periods (.), underscores (_), and hyphens (-). It must start with a letter or Chinese character but cannot start with `http:// or https://` .
Description	Enter a description for the physical connection. The description must be 2 to 256 characters in length. It must start with a letter or Chinese character but cannot start with `http://` or `https://` .
Peer Address	Enter the address of the data center to which you want to establish the physical connection.
Physical Connection Configurations	

Parameter	Description
Endpoint	Select the access point to which you want to connect the data center. Access points are Alibaba Cloud data centers located in different regions. Each region contains one or more access points. Different access points allow you to connect to Alibaba Cloud from different geolocations and support different connection types.
Port Type	Select 10 GE Single-mode Optical Port for this parameter.
Access Device	Select the device to be connected to the physical connection.
Physical Connection Bandwidth	Enter a value as the maximum bandwidth of the physical connection port. Unit: Mbit/s. Minimum value: 2.

5. Return to the **Exclusive Physical Connection** page, confirm that the physical connection is in the **Allocating** state.

Step 3: Enable the physical connection

To enable a physical connection, perform the following operations:

1. After resources are allocated for the physical connection, the physical connection changes to the Pending state. Click **Confirm** in the Actions column.
2. Refresh the **Exclusive Physical Connection** page. The state of the physical connection changes to **Enabled**.

Step 4: Create a VBR

After a physical connection is enabled, you must create a VBR to connect the data center to Alibaba Cloud.

To create a VBR, perform the following operations:

1. Log on to the Express Connect console.
2. In the left-side navigation pane, click **Virtual Border Routers (VBRs)**.
3. On the **Virtual Border Routers (VBRs)** page, click **Create VBR**.
4. Set the following parameters and click **OK** to create a VBR.

Parameter	Description
Account	◦ **Current account**: Create a VBR for the account that you use to log on to the console. ◦ **For others account create**: Create a VBR for another account. In this example, **Current account** is selected.
Account	This parameter is displayed only when the account type is set to **For others account create**. Enter the ID of the account for which you want to create the VBR.

Parameter	Description
Name	Enter a name for the VBR. The name must be 2 to 128 characters in length and can contain digits, periods (.), underscores (_), and hyphens (-). It must start with a letter but cannot start with `http:// or` `https://`. This parameter is displayed only when the account type is set to **Current account**.
Physical Connection Interface	Select a physical connection port that you want to associate with the VBR. The physical connection port must be in the Installed and Running state.
VLAN ID	Enter the VLAN ID of the VBR. Valid values: 0 to 2999. ○ If the VLAN ID is set to 0, the switch port of the VBR uses the Layer 3 router interface mode instead of the VLAN mode. In Layer 3 router interface mode, each physical connection is associated with a VBR. ○ If the VLAN ID is set to a value from 1 to 2999, the switch port of the VBR uses VLAN Layer 3 subinterface mode. In Layer 3 subinterface mode, each VLAN ID corresponds to a VBR. In this mode, the physical connection to which the VBR is established can connect to virtual private clouds (VPCs) that are created by different accounts. VBRs that correspond to different VLANs are isolated from each other at Layer 2. **Notice** The VLAN ID that you specify for the VBR and that of the customer-premises device must be the same.
Gateway IP Address on Alibaba Cloud Side	Enter the IPv4 address of the gateway that is deployed on Alibaba Cloud. In this example, `192.168.101.5` is entered.
Gateway IP Address on Customer Side	Enter the IPv4 address of the gateway that is deployed in the data center. In this example, `192.168.101.6` is entered.
Subnet Mask	Enter the subnet mask of the IPv4 addresses on the Alibaba Cloud and the data center sides. In this example, `255.255.255.252` is entered. You can enter a longer subnet mask because only two IP addresses are required.

5. When the VBR is in the **Active** state, **ping** the IP address of the gateway that is deployed on Alibaba Cloud from the gateway that is deployed in the data center to test the connectivity.

Step 5: Create a peering connection between the VBR and a VPC

After you associate the physical connection with the VBR, you must create a peering connection between the VBR and a VPC. This way, the VPC and the data center can communicate with each other through private connections.

To create a peering connection between the VBR and a VPC, perform the following operations:

1. Log on to the Express Connect console.
2. In the left-side navigation pane, choose **VPC Peering Connections > VBR-to-VPC**.
3. On the **VBR-to-VPC** page, click **Create Peering Connection**.
4. Set the following parameters and click **Submit** to configure the peering connection.

Parameter	Description
Scenarios	Use the default setting **VBR-to-VPC** for this parameter.
Common Settings	
Organization	Select the organization to which the VBR belongs.
Resource Set	Select the resource group to which the VBR belongs.
VBR-side Configurations	
Router Type	Use the default setting **VBR** for this parameter.
Region	Select the region where the VBR is created.
Endpoint	Select an access point.
VBR ID	Select the ID of the VBR.
Router Interface Specifications	Select the interface specification of the router. In this example, 1,000 Mbit/s is selected.
Router Interface Name	Enter a name for the router interface. The name must be 2 to 128 characters in length and can contain digits, periods (.), underscores (_), and hyphens (-). It must start with a letter but cannot start with `http:// or https://`.
Description	Enter a description for the router. The description must be 2 to 256 characters in length. It must start with a letter but cannot start with `http:// or https://`.
VPC-side Configurations	
Router Type	Use the default setting **vRouter** for this parameter.
Region	Select the region where the VPC is created.
VPC ID	Select the ID of the VPC.
Router Interface Name	Enter a name for the router interface. The name must be 2 to 128 characters in length and can contain digits, periods (.), underscores (_), and hyphens (-). It must start with a letter but cannot start with `http:// or https://`.
Health Check Source IP Address	Enter the source IP address of health checks. **Note** The source IP address must be an idle IP address that belongs to the CIDR block of a vSwitch in the VPC.

Parameter	Description
Health Check Destination IP Address	Enter the destination IP address of health checks. **Note** The destination IP address is the IP address of the customer-premises device that is connected to the VBR to which the physical connection is established.
Description	Enter a description for the VPC. The description must be 2 to 256 characters in length. It must start with a letter or Chinese character but cannot start with `http://` or `https://`.

5. Click **Submit**.

26.1.4.3. Delete a physical connection

This topic describes how to delete a physical connection that you no longer need.

To delete a physical connection that is established to a virtual private cloud (VPC) and a virtual border router (VBR), perform the following operations:

1. Delete routes on the router of the VPC and routes on the VBR.
2. If Border Gateway Protocol (BGP) routing is configured, you must delete the BGP peers and BGP groups.
3. Delete the connection between the VPC and the VBR.
4. Delete all VBRs that are associated with the physical connection.
5. Delete the physical connection.

Step 1: Delete routes

To delete routes on the router of a VPC and routes on a VBR, perform the following operations:

1. Log on to the VPC console.
2. In the left-side navigation pane, click **Route Tables**.
3. In the top navigation bar, select the region to which the route table belongs.
4. On the **Route Tables** page, find the route table that you want to manage and click **Manage** in the **Actions** column.
5. On the **Route Entry List** tab, find the custom route that you want to delete and click **Delete** in the **Actions** column.
6. In the **Delete Route Entry** message, click **OK**.
7. Log on to the Express Connect console.
8. In the left-side navigation pane, click **Virtual Border Routers (VBRs)**.
9. On the **Virtual Border Routers (VBRs)** page, select the region where the VBR that you want to manage is created and click the ID of the VBR.
10. On the **Routes** tab, find the route that you added and click **Delete** in the **Actions** column.
11. In the **Delete Route** message, click **OK**.

Step 2: Delete BGP peers and BGP groups

If BGP is configured, you must perform the following operations to delete the VBR-related BGP settings:

1. Log on to the Express Connect console.
2. In the left-side navigation pane, click **Virtual Border Routers (VBRs)**.

3. On the **Virtual Border Routers (VBRs)** page, select the region where the VBR that you want to manage is created and click the ID of the VBR.
4. On the **BGP Peers** tab, delete the BGP peers that you added.
5. On the **BGP Groups** tab, delete the BGP groups that you created.
6. On the **Advertised BGP Subnets** tab, delete the BGP CIDR blocks that are advertised.

Step 3: Delete the connection between the VBR and the VPC

To delete the connection between the VPC and the VBR, perform the following operations:

1. Log on to the CEN console.
2. On the **Instances** page, find the target CEN instance, and then click **Manage** in the **Actions** column.
3. On the **Networks** tab page, find the target network instance, and then click **Detach** in the **Actions** column.
4. In the **Detach Network** dialog box, click **OK**.

Step 4: Delete a VBR

To delete a VBR, perform the following operations:

1. Log on to the Express Connect console.
2. In the left-side navigation pane, click **Virtual Border Routers (VBRs)**.
3. On the **Virtual Border Routers (VBRs)** page, find the VBR that you want to delete and click **Delete** in the **Actions** column.
4. In the **Delete VBR** message, click **Confirm**.

Step 5: Delete a physical connection

To delete a physical connection, perform the following operations:

1. Log on to the Express Connect console.
2. In the left-side navigation pane, click **Exclusive Physical Connection**.
3. On the **Exclusive Physical Connection** page, find the physical connection that you want to delete and click **Terminate Connection**.
4. In the **Terminate Connection** message, click **Confirm**.
5. On the **Exclusive Physical Connection** page, find the physical connection that you want to delete and click **Delete** in the Actions column.
6. In the **Delete Physical Connection** message, click **Confirm**.

26.1.5. VBRs

26.1.5.1. What is a VBR?

Virtual border routers (VBRs) are virtualization of physical connection ports that are isolated by Alibaba Cloud. VBRs are defined on top of the Layer 3 overlay and switch virtualization technologies in the Software Defined Network (SDN) architecture. A VBR functions as a router between a customer-premises device and a virtual private cloud (VPC). VBRs route traffic between VPCs and data centers.

Each VBR is associated with a route table, which is similar to the vRouter of a VPC. You can add routes to the route table of a VBR to control traffic forwarding on the VBR.

Features

VBRs provide the following features:

- Exchange data between a VPC and a data center.

- Add or identify VLAN tags in Layer 3 subinterface mode.
- Determine the mode of a physical connection port: Layer 3 router interface or VLAN Layer 3 subinterface.
- Support Border Gateway Protocol (BGP).

  BGP is a dynamic routing protocol based on Transmission Control Protocol (TCP). BGP is used to exchange routing information and network accessibility information among autonomous systems. When you create physical connections to connect a data center to a VBR, you can configure BGP settings. BGP helps you build hybrid clouds in a more efficient, flexible, and reliable way.

Limits

- Source address-specific policy-based routes are not supported.
- Each VBR can be associated with only one route table.
- VBRs support only BGP4.
- VBRs support IPv4 BGP. IPv6 BGP is not supported.
- You can create up to eight BGP peers for each VBR.
- Each BGP peer supports up to 100 dynamic routes. If the number of dynamic routes that a BGP peer learns reaches the upper limit, the system generates an alert. However, the dynamic routes are not denied.
- To configure BGP when you connect to a VPC, you must specify an Autonomous System Number (ASN) for the VPC. The ASN that you specify must not be the same as the ASNs of the vSwitches in the VPC.

26.1.5.2. Create a VBR

After a physical connection is enabled, you must create a virtual border router (VBR) for the physical connection. The VBR is used to route traffic between the virtual private cloud (VPC) and data center that are connected through the physical connection.

Context

A VBR is a router deployed between a VPC and a customer-premises equipment (CPE) in a data center. Each VBR is associated with a route table. To manage traffic forwarding on a VBR, you can add routes to the route table that is associated with the VBR.

A VBR provides the following features:

- Exchanges data as a router between a VPC and a data center.
- Determines the port mode of the physical connection: Layer 3 router interface mode or VLAN Layer 3 subinterface mode.
- Adds or identifies VLAN tags in Layer 3 subinterface mode.
- Supports Border Gateway Protocol (BGP) dynamic routing.

Procedure

1. Log on to the Express Connect console.
2. In the left-side navigation pane, click **Virtual Border Routers (VBRs)**.
3. On the **Virtual Border Routers (VBRs)** page, click **Create VBR**.
4. Configure the VBR based on the following information and click **OK**.

Parameter	Description
Account	Select the type of account to which the VBR that you want to create belongs: **Current account** or **For others account create**.

Parameter	Description
Account	Enter the primary key of the account. This parameter is displayed only when you select **For others account create**. **Note** To obtain the primary key of the account, log on to the Apsara Uni-manager Management Console. In the top navigation bar, choose **Enterprise > Organizations > Obtain an accesskey**.
Name	Enter a name for the VBR. The name must be 2 to 128 characters in length, and start with a letter or Chinese character. It can contain digits, periods (.), underscores (_), and hyphens (-). It cannot start with `http://` or `https://`. This parameter is displayed only when the account type is set to **Current account**.
Physical Connection Interface	Select a physical connection port to associate with the VBR. The physical connection port must be installed and function as expected.
VLAN ID	The VLAN ID of the VBR. Valid values: 0 to 2999. ◦ If the VLAN ID is set to 0, the switch port of the VBR uses the Layer 3 router interface mode instead of the VLAN mode. In Layer 3 router interface mode, each physical connection is associated with a VBR. ◦ If the VLAN ID is set to a value from 1 to 2999, the switch port on the VBR uses VLAN Layer 3 subinterface mode. In Layer 3 subinterface mode, each VLAN ID corresponds to a VBR. In this mode, the physical connection that is established to the VBR can connect to VPCs created by different accounts. VBRs that correspond to different VLANs are isolated from each other at Layer 2. For example, assume that a company has multiple subdivisions or subsidiaries. Each subdivision or subsidiary has a separate Alibaba Cloud account. Each account has a separate VPC. If the company applies for a physical connection, they must assign a VLAN ID for each subdivision or subsidiary. When the company creates router interfaces, they use VLAN IDs to identify the subsidiaries or subdivisions that use the physical connection. This way, the subsidiaries or subdivisions are isolated at Layer 2.
Gateway IP Address on Alibaba Cloud Side	Enter the IP address of the gateway that routes traffic from the VPC to the data center.
Gateway IP Address on Customer Side	Enter the IP address of the gateway that routes traffic from the data center to the VPC.
Subnet Mask	Enter the subnet mask of the gateway IPv4 address on the Alibaba Cloud side and the gateway IPv4 address on the customer side. You can enter a longer subnet mask because only two IP addresses are required.

26.1.5.3. Configure BGP

This topic describes how to establish Border Gateway Protocol (BGP) peering relationships between a data center and a virtual border router (VBR). To perform this task, you must add the BGP peer that communicates with the VBR to a BGP group, and then advertise the BGP CIDR block on the VBR.

> **Note** Express Connect allows you to establish BGP peering relationships only between a VBR and a data center. However, when you establish a physical connection, you still need to add two routes on the VBR. You must add a route that points to the physical connection and a route that points to the virtual private cloud (VPC). For more information, see Add routes.

What is BGP?

BGP is a dynamic routing protocol based on Transmission Control Protocol (TCP). BGP is used to exchange routing information and network accessibility information in different autonomous systems. When you create a physical connection to connect a data center to a VBR, you can configure BGP routing to enable private communication. BGP can help you build hybrid clouds in a more efficient, flexible, and reliable manner.

Before you configure BGP, you must create a BGP group. BGP groups are used to simplify BGP configurations. You can save time and effort by adding BGP peers that use the same configurations to one BGP group. This way, you only need to create a BGP group with an Autonomous System Numbers (ASN) and add BGP peers that meet your requirements to the BGP group. After you add the BGP peers to the BGP group, the BGP peers share the configurations of the BGP group. This saves you the need to configure each BGP peer separately.

Limits

BGP has the following limits:

- VBRs can establish peering relationships only with data centers that are connected to the VBRs through physical connections. Static routing is still required between the VBRs and VPCs.
- VBRs support only BGP 4.
- You can create up to eight BGP peers with each VBR.
- Each BGP peer supports at most 110 dynamic routes. If the number of dynamic routes that a BGP peer learns reaches the upper limit, the system generates an alert. However, the dynamic routes are not denied.
- To configure BGP when you connect to a VPC, you must specify an ASN for the VPC. The ASN that you specify must not be the same as the ASNs of the vSwitches in the VPC.

Step 1: Create a BGP group

Before you configure BGP routing, you must create a BGP group with the requested ASN.

To create a BGP group, perform the following operations:

1. Log on to the Express Connect console.
2. In the left-side navigation pane, select **Virtual Border Routers (VBRs)**.
3. On the **Virtual Border Routers (VBRs)** page, click the ID of the VBR that you want to manage.
4. Click the **BGP Groups** tab and click **Create BGP Group**.
5. Set the following parameters to configure the BGP group and click **OK**.

Parameter	Description
Name	Enter a name for the BGP group. The name must be 2 to 128 characters in length, and can contain digits, periods (.), underscores (_), and hyphens (-). The name must start with a letter or Chinese character. It cannot start with `http://` or `https://`.
Peer ASN	Enter the ASN of the data center network.
BGP Key	Enter the key of the BGP group.
Description	Enter a description for the BGP group. The description must be 2 to 256 characters in length. It must start with a letter but cannot start with `http://` or `https://`.

Parameter	Description
Local AS	Enter the ASN of the VPC.

Step 2: Add a BGP peer

To add a BGP peer, perform the following operations:

1. Log on to the Express Connect console.
2. In the left-side navigation pane, click **Virtual Border Routers (VBRs)**.
3. On the **Virtual Border Routers (VBRs)** page, click the ID of the VBR that you want to manage.
4. Click the **BGP Peers** tab and click **Create BGP Peer**.
5. Configure the BGP peer and click **OK**.

Parameter	Description
BGP Group	Select the BGP group to which you want to add the BGP peer.
BGP Peer IP Address	Enter the IP address of the BGP peer.

6. Return to the **Virtual Border Routers (VBRs)** page and click the **BGP Peers** tab to check the state of the BGP peer.

   A BGP peer can be in the following states:

   - **Idle**: indicates that the BGP peer is idle. Idle is the initial state of a BGP peering relationship. In this state, BGP waits for a start event. After the start event occurs, BGP initializes all resources and resets the ConnectRetry timer. Then, BGP initiates a TCP connection while the BGP peer changes to the Connect state.
   - **Connect**: indicates that a connection is being established to the BGP peer. In the Connect state, BGP initiates the first TCP connection request. If the ConnectRetry timer depletes before the TCP connection is established, a new TCP connection request is initiated and the BGP peer remains in the Connect state.
     - If the TCP connection is established, the BGP peer changes to the OpenSent state.
     - If the TCP connection attempt fails, the BGP peer changes to the Active state.
   - **Active**: indicates that the BGP peer is active. In the Active state, BGP attempts to establish the TCP connection again. If the ConnectRetry timer depletes, the BGP peer switches back to the Connect state.
     - If the TCP connection is established, the BGP peer changes to the OpenSent state.
     - If the TCP connection attempt fails, the BGP peer remains in the Active state and BGP continues to initiate TCP connection requests.
   - **OpenSent**: indicates that an OPEN message is sent. The OpenSent state indicates that the TCP connection is established. The first OPEN message has been sent to the BGP peer. BGP is waiting for an OPEN message from the BGP peer. After BGP receives the OPEN message, it checks the message for errors.
     - If the message contains errors, the system returns an error message to the BGP peer and the BGP peer switches back to the Idle state.
     - If the OPEN message does not contain any errors, BGP sends a Keepalive message and resets the Keepalive timer. In this case, the BGP peer changes to the OpenConfirm state.
   - **OpenConfirm**: indicates that BGP has confirmed the OPEN message. In the OpenConfirm state, BGP sends a Keepalive message and resets the timer.
     - If the Keepalive message is received before the timer expires, the BGP peer changes to the Established state. This state indicates that the BGP peering relationship is established.
     - If the TCP connection is interrupted, the BGP peer switches back to the Idle state.

- **Established**: indicates that the BGP peering relationship is established. In this state, the VBR and the BGP peer send Update messages to each other and the timer is reset.
- **UnEstablished**: indicates that the BGP peering relationship is not established.

Step 3: Advertise the BGP CIDR block

After you configure BGP peers, you must advertise the CIDR block of the VPC to complete the BGP configuration. After the BGP peering relationship is established, the VBR automatically learns the CIDR block of the data center.

To advertise the CIDR block of the VPC, perform the following operations:

1. Log on to the Express Connect console.
2. In the left-side navigation pane, click **Virtual Border Routers (VBRs)**.
3. On the **Virtual Border Routers (VBRs)** page, click the ID of the VBR that you want to manage.
4. Click the **Advertised BGP Subnets** tab and click **Advertise BGP Subnet**.
5. Enter the CIDR block that you want to advertise and click **OK**.

Step 4: Configure BGP for the data center

The preceding steps describe how to configure BGP on Alibaba Cloud. After you configure BGP on Alibaba Cloud, you must configure BGP for the data center and advertise routes of the network device in the data center. For more information about the command, consult the manufacturer of the network device.

26.1.5.4. Add routes

This topic describes how to add routes to the route table of a virtual border router (VBR). You can configure routes to manage where traffic is routed on a VBR.

Context

To forward network traffic between a virtual private cloud (VPC) and a data center, you must add two routes to the route table of the VBR that is used to connect the data center to the VPC. One of the routes points to the physical connection and the other route points to the VPC. You can configure BGP routing for a data center on a VBR. For more information, see Configure BGP.

Take note of the following items when you manage routes in the route table of a VBR:

> Notice
> - You can query routes by type. Fuzzy match is supported.
> - Each route table supports up to 48 custom routes.
> - Source address-specific policy-based routes are not supported.

Procedure

1. Log on to the Express Connect console.
2. In the left-side navigation pane, click **Virtual Border Routers (VBRs)**.
3. On the **Virtual Border Routers (VBRs)** page, click the ID of the VBR that you want to manage.
4. Click the **Routes** tab and click **Add Route**.
5. Configure the route and click **OK**.

Parameter	Description

Parameter	Description
Next Hop Type	The type of next hop. You can select the following types: ○ **VPC**: The VBR routes traffic to a VPC based on the destination CIDR block. ○ **Physical Connection Interface**: The VBR routes traffic to a physical connection port based on the destination CIDR block.
Destination CIDR block	Enter the destination CIDR block.
Next Hop	Select the next hop based on the specified type.

26.1.5.5. Create a peering connection

This topic describes how to create a peering connection. After you create a virtual border router (VBR), you must create a peering connection between the VBR and a virtual private cloud (VPC). This way, the VBR can route traffic between the VPC and the data center that is connected to the VBR.

For more information about how to create a peering connection, see Connect two VPCs.

27.Apsara Stack Security

27.1. User Guide

27.1.1. What is Apsara Stack Security

Apsara Stack Security is a solution that protects Apsara Stack assets with a full suite of security features, such as network, server, application, data, and security management.

Background information

Traditional security solutions for IT services detect attacks on network perimeters. These solutions use hardware products such as firewalls and intrusion prevention systems (IPSs) to protect networks against attacks.

With the development of cloud computing, an increasing number of enterprises and organizations use cloud computing services instead of traditional IT services. Cloud computing features low costs, on-demand flexible configuration, and high resource utilization. Cloud computing environments do not have definite network perimeters. As a result, traditional security solutions cannot effectively safeguard cloud assets.

With the powerful data analysis capabilities and professional security operations team of Alibaba Cloud, Apsara Stack Security provides integrated security protection services for networks, applications, and servers.

Complete security solution

Apsara Stack Security consists of Apsara Stack Security Standard Edition and optional security services to provide a comprehensive security solution.

Security domain	Service name	Description
Security management	Threat Detection Service (TDS)	Monitors traffic and overall security status to audit and centrally manage assets.
Server security	Server Guard	Protects Elastic Compute Service (ECS) instances against intrusions and malicious code.
	Server Security	Protects physical servers against intrusions.
Application security	Web Application Firewall (WAF)	Protects web applications against attacks and ensures that mobile and PC users can securely access web applications over the Internet.
Network security	Anti-DDoS	Ensures the availability of network links and improves business continuity.
Data security	Sensitive Data Discovery and Protection (SDDP)	Prevents data leaks and helps your business system meet compliance requirements.
Security O&M service	On-premises security service	Helps you establish and optimize the cloud security system to protect your business system against attacks by using security features of Apsara Stack Security and other Apsara Stack services.

27.1.2. Precautions

Before you log on to the Apsara Stack Security console, you must verify that your local PC meets the configuration requirements.

The configuration requirements for the local PC are listed in Configuration requirements.

Configuration requirements

Item	Requirement
Browser	• Internet Explorer: V11 or later • Chrome (recommended): V42.0.0 or later • Firefox: V30 or later • Safari: V9.0.2 or later
Operating system	• Windows XP, Windows 7, or later • macOS

27.1.3. Quick start

27.1.3.1. User roles and permissions

This topic describes the user roles involved in Apsara Stack Security.

All roles in Apsara Stack Security Center are provided by default. You cannot add custom roles. Before you log on to Apsara Stack Security Center, make sure that your account is assigned the required role. For more information, see Default roles in Apsara Stack Security.

Default roles in Apsara Stack Security

Role	Permission
System administrator of Apsara Stack Security Center	Manages and configures system settings for Apsara Stack Security Center. The system administrator has permissions to manage Apsara Stack accounts, synchronize data, configure alerts, and configure global settings.
Security administrator of Apsara Stack Security Center	Monitors the security status across Apsara Stack and configures security policies for each functional module of Apsara Stack Security. The security administrator has permissions on all features under Threat Detection, Network Security, Application Security, Server Security, PM Security, and Asset Management. **Note** The permissions on Web Application Firewall (WAF) and Cloud Firewall must be separately assigned.
Department security administrator	Monitors the security status of cloud resources in a specific department and configures security policies for each functional module of Apsara Stack Security for this department. The department security administrator has permissions on all features under Threat Detection, Network Security, Application Security, Server Security, PM Security, and Asset Management. In addition, the department security administrator can specify alert notification methods and alert recipients in the department. **Note** The permissions on WAF and Cloud Firewall must be separately assigned.
Auditor of Apsara Stack Security Center	Conducts security audits across Apsara Stack. The auditor can view audit events and original logs, configure audit policies, and access all features under Security Audit.

If you do not have an account that assumes the required role, contact the administrator to create an account and assign the role to the account. For more information, see the **Create a user** topic in the *Apsara Uni-manager Management Console User Guide*.

27.1.3.2. Log on to Apsara Stack Security Center

This topic describes how to log on to Apsara Stack Security Center.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Security > Apsara Stack Security**.
5. On the **Apsara Stack Security Center** page, select **Region**.
6. Click **YD** to go to Apsara Stack Security Center.

27.1.4. Threat Detection Service

27.1.4.1. Threat Detection Service overview

This topic describes the basic concepts of Threat Detection Service (TDS).

TDS provides comprehensive protection for enterprises. It can monitor vulnerabilities, intrusions, web attacks, DDoS attacks, threat intelligence, and public opinions. TDS uses modeling and analysis to obtain key information based on traffic characteristics, host processes, and host operations logs. In addition, TDS identifies intrusions that cannot be detected by traffic inspection or file scan. You can use the input of cloud analysis models and intelligence data to discover sources and behavior of attacks and assess threats.

TDS provides the following features:

- Overview: provides a security situation overview and information about security screens.
- Security Alerts: displays security alerts that occur in the business system.
- Attack Analysis: displays application attacks and brute-force attacks that occur in the system.
- Cloud Service Check: checks whether the security configuration for cloud services has risks.
- Application Whitelist: provides information about application processes on servers that require protection

based on intelligent learning. The application processes are identified as trusted, suspicious, or malicious. This prevents unauthorized processes.

- Assets: manages servers and cloud services on Apsara Stack.
- Security Reports: allows you to configure security report tasks on Apsara Stack.

27.1.4.2. Security overview

27.1.4.2.1. View security overview information

This topic describes how to view security statistics, attack trends, and network traffic information on the Apsara Stack platform.

Context

The **Security Overview** tab provides an overview of detected security events, the latest threats, and inherent vulnerabilities of the system. A security administrator can view information on the **Security Overview** tab to better understand the security posture of the system.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Threat Detection > Overview**. By default, the **Security Overview** tab appears.
3. View the security posture of the Apsara Stack platform.

   Sections on the Security Overview tab

Section	Description
Secure Score	The security score of assets and the number of detected security risks.
Asset Status	The total number of assets and the numbers of servers that are not protected, servers that are stopped, and servers that are at risk.
Security Detection And Defense Capabilities	The numbers of precise defense events and anti-tampering events over the last 15 days, the time when the antivirus database was last updated, and the time when vulnerability scanning was last performed. This allows you to obtain the defense situation and security status of your assets in real time.
Threat statistics	The numbers of alerts that are not handled, vulnerabilities that are not fixed, baseline risks, and attacks.
Config Assessment Risks	Risks in the baseline configurations of cloud services.
Issue Resolved	Statistics on alerts, vulnerabilities, and baseline risks that have been processed over the last 15 days. The statistics are displayed in a bar and trend chart.

27.1.4.3. Security alerts

27.1.4.3.1. View security alerts

This topic describes how to view security alerts on the Security Alerts page.

Procedure

1.

2. In the left-side navigation pane, choose **Threat Detection > Security Alerts**.
3. (Optional)Set filter conditions for security alerts.

> **Note** If you want to view all alerts, do not set the conditions.

Filter condition	Description
Severity	The severity level. You can select one or more levels. Valid values: ○ Urgent ○ Warning ○ Notice
Alert status	The status of alerts. Valid values: ○ Unhandled Alerts ○ Handled
Alert type	The type of alerts. Select **All** or a specific type.
Affected asset group	The affected asset group. Select **Asset Group** or a specific group.
Search for alerts by name or asset	Enter an alert name or a keyword of affected assets to search for alerts.

4. View security alerts and their details in the list.

27.1.4.3.2. Manage quarantined files

This topic describes how to manage threat files that are quarantined by the system. The system deletes a quarantined file 30 days after the file is quarantined. You can restore the file before it is deleted.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Security Alerts**.
3. In the upper-right corner of the **Alerts** page, click **Quarantine**.
4. In the **Quarantine** panel, view the information about a quarantined file, such as the host, path, status, and operation time.
5. (Optional)If a file is incorrectly quarantined, click **Restore** in the **Actions** column to restore the file.

> **Notice** Before you can restore a quarantined file, make sure that the file is normal and does not bring risks.

The restored file is removed from the quarantine and is displayed in the security alert list again.

27.1.4.3.3. Configure security alerts

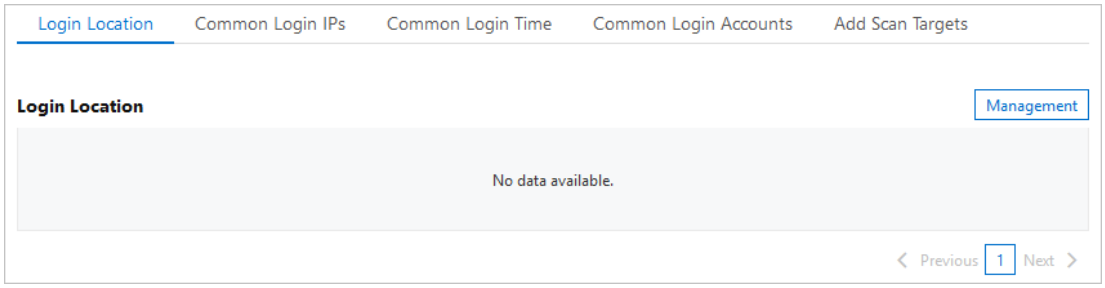
This topic describes how to configure logon settings and web directories that are scanned.

Procedure

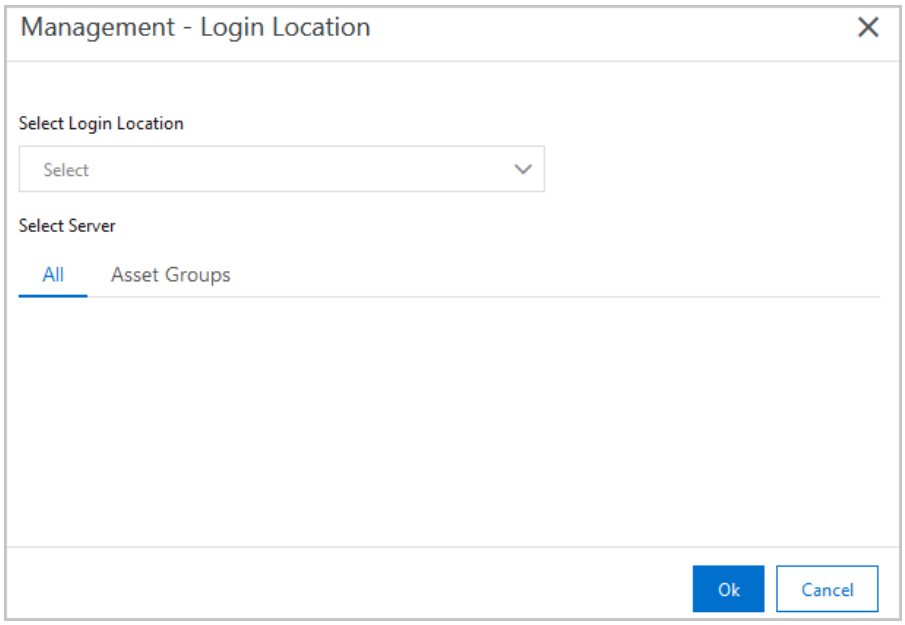
1.
2. In the left-side navigation pane, choose **Threat Detection > Security Alerts**.
3. In the upper-right corner of the page, click **Settings**.

   You can perform the following operations:

   - **Manage a common logon location.**
     a. On the right side of **Login Location**, click **Add**.



     b. Select a logon location and the servers that are allowed to be logged on to from the location.



     c. Click **OK**.

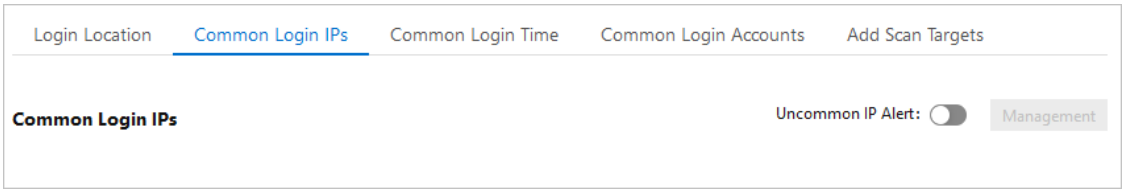
   Threat Detection Service (TDS) allows you to **edit** and **delete** added logon locations.

   - Find the target logon location and click **Edit** on the right side to change the servers that are allowed to be logged on to from this location.
   - Find the target logon location and click **Delete** on the right side to delete the logon location.

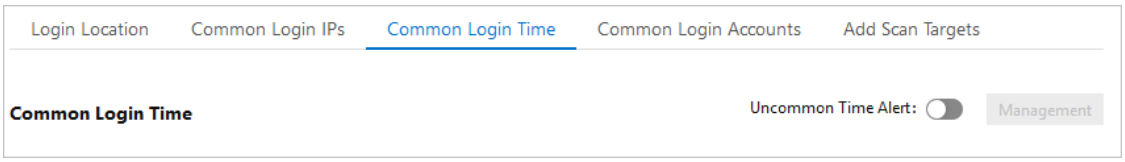
   - **Configure advanced logon settings.**

> **Note** You can specify the IP addresses, accounts, and time periods that are allowed to log on to your assets. After you configure these settings, alerts are triggered if your assets receive logon requests that do not meet the requirements. The procedure to configure advanced logon settings is similar to that to configure **common logon locations**. You can follow the preceding procedure to **add**, **edit**, and **delete** advanced logon settings.

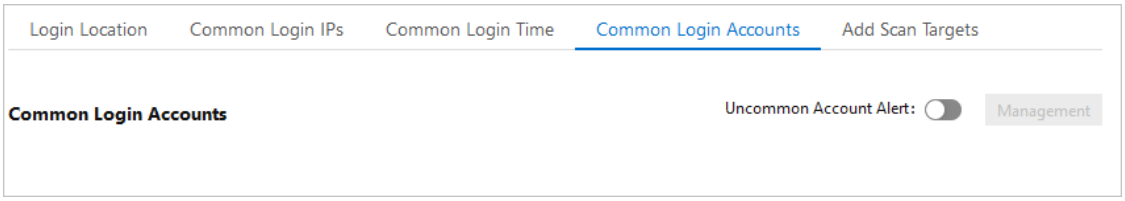
- On the right side of **Common Login IPs**, enable or disable Uncommon IP Alert. After this function is enabled, alerts are triggered if your assets receive logon requests from unauthorized IP addresses.



- On the right side of **Common Logon Time**, enable or disable Uncommon Time Alert. After this function is enabled, alerts are triggered if your assets receive logon requests at unauthorized time.

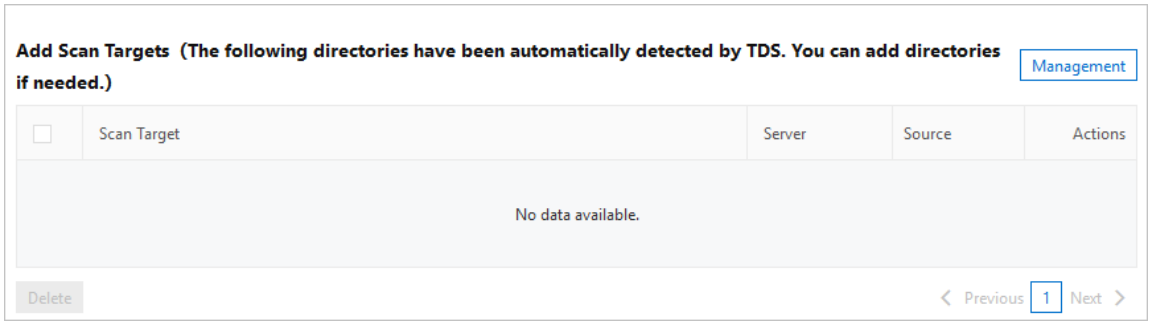


- On the right side of **Common Login Accounts**, enable or disable Uncommon Account Alert. After this function is enabled, alerts are triggered if your assets receive logon requests from unauthorized accounts.



- **Add a web directory as the scan target.**

  TDS automatically detects web directories on your servers.



  It performs dynamic and static scans on these directories. You can also manually add web directories for scanning.

  a. On the right side of **Add Scan Targets**, click **Add**.

  b. Enter a valid web directory and select the servers on which the directory is scanned. The web directory is added to the scan list.

  > **Note** To ensure performance and efficiency, do not enter a root directory.

  c. Click **OK**.

27.1.4.4. Attack analysis

This topic describes the statistics provided by the attack analysis feature. The statistics include the total number of attacks, distribution of attack types, top five attack sources, top five assets attacked, and an attack list.

Background information

The attack analysis feature provides basic attack detection and prevention capabilities in Apsara Stack Security Center. We recommend that you optimize firewalls and enhance business security to develop a more fine-grained and in-depth defense system.

On the **Attack Awareness** page, you can specify a time range to view these attack details. You can view the attack analysis statistics of the current day, last 7 days, or last 30 days. You can also set Time Range to **Custom** to view the statistics of a time range within the last 30 days.

- **Attacks**: the total number of attacks detected in your assets within a specific time range.
- **Attack Type Distribution**: the attack types and the number of attacks for each type.
- **Top 5 Attack Sources**: the top five IP addresses from which the most attacks are launched.
- **Top 5 Attacked Assets**: the top five assets that are attacked the most frequently.
- **Attack list**: the details about each attack. The details include the attack time, source IP address, attacked asset, attack type, and attack status.

  > **Note** The attack list displays a maximum of 10,000 attacks. You can specify **Time Range** to view details about the attacks that occur over the specified time range.

  Parameters in the attack list

Parameter	Description
Attacked At	The time at which an attack occurs.
Attack Source	The source IP address of an attack.
Attacked Asset	The name, public IP address, and private IP address of an attacked asset.
Attack Method	The HTTP request method used to initiate an attack. Valid values: POST and GET.
Attack Type	The type of an attack. The types of attacks that can be detected include SSH brute-force attacks and remote code execution attacks.

- Search for an attack.

  To view the details about a specific attack, specify search conditions in the search box above the attack list. Search conditions include the attack type, attacked asset, and source IP address.

- View the details of an attacked asset.

  To view the details about an attacked asset, move the pointer over the name of the attacked asset.

- Export the attack list.

  To export and save the attack list to your computer, you can click the  icon in the upper-left corner above the attack list. The attack list is exported to an Excel file.

27.1.4.5. Cloud service check

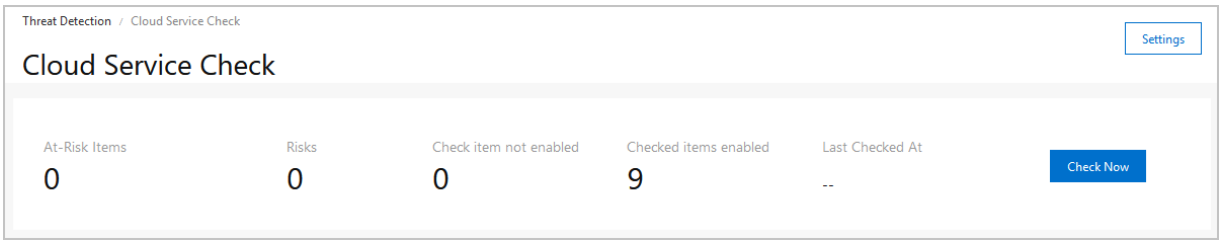
27.1.4.5.1. Overview

Threat Detection Service (TDS) provides the cloud service check feature. This feature allows you to check for security risks in the configurations of your Apsara Stack services. This topic describes the features and check items that are supported by the cloud service check feature.

Background information

The cloud service check feature allows you to perform network access control and data security checks. The checks help you detect configuration risks of your Apsara Stack services and provide repair solutions.

You can view the number of **Checked items enabled** on the **Cloud Service Check** page.



Cloud service check list

The following table describes the check items.

Type	Supported item	Description
	PolarDB - Backup configurations	Checks whether the automatic backup feature is enabled for PolarDB. Regular backups help you improve database security. If an error occurs in your database after backup is performed, you can restore data. PolarDB supports automatic backup. We recommend that you enable automatic backup to create a backup on a daily basis.
	Container Registry - Repository permission configurations	Checks whether the status of Container Registry repositories is set to private. Container Registry repositories include public repositories and private repositories. Public repositories allow Internet users to anonymously download information. If an image contains sensitive information, we recommend that you set the status of Container Registry repositories to private. Otherwise, ignore related alerts.
	OSS - Server-side encryption	Checks whether the data encryption feature is enabled for Object Storage Service (OSS) buckets. OSS supports server-side encryption to ensure the security of data that is persistently stored in OSS. We recommend that you enable server-side encryption to protect sensitive data.
	OSS - Sensitive information leakage scans	Checks whether OSS sensitive files require access permissions.
	ApsaraDB RDS - Cross-region backup configurations	Checks whether cross-region backup is enabled for ApsaraDB RDS instances. ApsaraDB RDS for MySQL provides the cross-region backup feature that automatically synchronizes local backup files to OSS in another region. We recommend that you enable the cross-region backup feature.
	KVStore for Redis - Backup configurations	Checks whether the data backup feature is enabled for KVStore for Redis instances.

Type	Supported item	Description
Data security	ApsaraDB for MongoDB - SSL encryption	Checks whether SSL encryption is enabled for ApsaraDB for MongoDB databases. We recommend that you enable the SSL encryption feature to improve the security of data links in ApsaraDB for MongoDB databases.
	ApsaraDB for MongoDB - Backup configurations	Checks whether the automatic backup feature is enabled for ApsaraDB for MongoDB databases. Regular backups help you improve database security. If an error occurs in your database after backup is performed, you can restore data. ApsaraDB for MongoDB provides automatic backup policies. We recommend that you enable automatic backup to create a backup on a daily basis.
	ECS - Disk encryption	Checks whether encryption is enabled for disks on Elastic Compute Service (ECS) instances.
	ECS - Automatic snapshot policies	Checks whether the automatic snapshot feature is enabled for the disks on ECS instances. The automatic snapshot feature improves the security of ECS data and supports disaster recovery.
	OSS - Bucket permissions	Checks whether the OSS bucket ACL is set to *private*.
	OSS - Logging	Checks whether the logging feature is enabled for OSS.
	OSS - Cross-region replication configurations	Checks whether the cross-region replication feature is enabled for OSS.
	ApsaraDB RDS - Database security policies	Checks whether the SQL audit, SSL encryption, and transparent database encryption features are enabled for ApsaraDB RDS databases.
	ApsaraDB RDS - Backup configurations	Checks whether the data backup feature is enabled for ApsaraDB RDS instances.
	SSL Certificates Service - Expiration check	Checks whether your SSL certificate is expired. If your SSL certificate is expired, you are not allowed to use SSL Certificates Service.
	ECS - Security group policies	Checks ECS security group policies. We recommend that you grant minimum permissions to users. We also recommend that you specify 0.0.0.0/0 only for public-facing ports, such as port 80, 443, 22, or 3389.
	OSS - Bucket hotlink protection	Checks whether hotlink protection is enabled for OSS buckets. The OSS hotlink protection feature checks the Referer header to deny access from unauthorized users. We recommend that you enable this feature.
	VPC - DNAT rules	Checks whether a port is open to the Internet. When you create a Destination Network Address Translation (DNAT) rule for a NAT Gateway that is deployed in a virtual private cloud (VPC), we recommend that you do not open internal management ports to the Internet. Do not open all ports or an important port, for example, ports 22, 3389, 1433, or 3306, to the Internet.

Type	Supported item	Description
Network access control	Apsara Stack Security - Back-to-origin configuration checks	Checks whether Anti-DDoS Pro or Anti-DDoS Premium is configured to allow only Web Application Firewall (WAF) back-to-origin IP addresses. After you set up Anti-DDoS Pro, Anti-DDoS Premium, or WAF, you must hide the IP addresses of the backend servers to prevent attacks on the cloud assets.
	Apsara Stack Security - WAF back-to-origin configurations	Checks whether WAF allows requests only from WAF back-to-origin IP addresses. After you set up Anti-DDoS Pro, Anti-DDoS Premium, or WAF, you must hide the IP addresses of the backend servers to prevent attacks on the cloud assets.
	SLB - IP address whitelist configurations	Checks the access control configurations of Server Load Balancer (SLB) instances. Checks whether access control is enabled for HTTP and HTTPS services and checks whether 0.0.0.0/0 is added to the IP address whitelist.
	SLB - Open ports	Checks whether ports of SLB instances are unnecessarily open to the Internet.
	ApsaraDB RDS - IP address whitelist configurations	Checks whether a whitelist is configured for ApsaraDB RDS and whether the whitelist contains 0.0.0.0/0. If the whitelist contains 0.0.0.0/0, requests from all IP addresses are allowed. We recommend that you configure the whitelist to allow requests only from specific IP addresses.
	KVStore for Redis - IP address whitelist configurations	Checks whether a whitelist is configured for KVStore for Redis and whether the whitelist contains 0.0.0.0/0. If the whitelist contains 0.0.0.0/0, requests from all IP addresses are allowed. We recommend that you configure the whitelist to allow requests only from specific IP addresses.
	AnalyticDB for PostgreSQL - IP address whitelist configurations	Checks whether a whitelist is configured for AnalyticDB for PostgreSQL and whether the whitelist contains 0.0.0.0/0. If the whitelist contains 0.0.0.0/0, requests from all IP addresses are allowed. We recommend that you configure the whitelist to allow requests only from specific IP addresses.
	PolarDB - IP address whitelist configurations	Checks whether a whitelist is configured for PolarDB to allow access from the Internet and whether the whitelist contains `0.0.0.0/0`. If the whitelist contains 0.0.0.0/0, requests from all IP addresses are allowed. We recommend that you configure the whitelist to allow requests only from specific IP addresses.
	ApsaraDB for MongoDB - IP address whitelist configurations	Checks whether a whitelist is configured for ApsaraDB for MongoDB and whether the whitelist contains `0.0.0.0/0`. If the whitelist contains 0.0.0.0/0, requests from all IP addresses are allowed. We recommend that you configure the whitelist to allow requests only from specific IP addresses.

27.1.4.5.2. Run cloud service checks

Threat Detection Service (TDS) provides the cloud service check feature. This feature allows you to check for security risks in the configurations of your cloud services. This topic describes how to manually run cloud service checks on your cloud services. This topic also describes how to specify a detection interval for automated periodic checks.

Context

Apsara Stack Security supports manual checks and automated periodic checks to scan for configuration risks in cloud services.

- Manual checks: On the **Cloud Service Check** page, you can click **Check Now** to check for security risks in the configurations of your cloud services.
- Automated periodic checks: By default, Apsara Stack Security automatically runs checks during the time range `00:00:00-06:00:00` every other day. You can also customize a time range for automated periodic checks. This way, you can detect and handle configuration risks at the earliest opportunity.

Manual checks

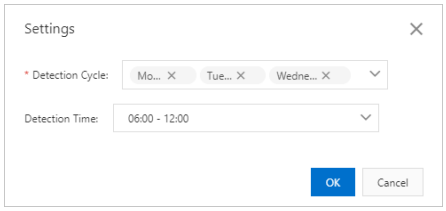
1.
2. In the left-side navigation pane, choose **Threat Detection > Cloud Service Check**.
3. On the **Cloud Service Check** page, click **Check Now** to check whether the configurations of all your cloud services contain risks and the number of affected assets.

   > **Note** Do not perform other operations until the check is complete.

   After the check is complete, the detected risks are listed based on their severities in descending order.

Automated checks

1.
2. In the left-side navigation pane, choose **Threat Detection > Cloud Service Check**.
3.
4. In the **Settings** dialog box, configure the **Detection Cycle** and **Detection Time** parameters.

   - **Detection Cycle**: supports Monday to Sunday. You can select multiple days of the week.
   - **Detection Time**: supports `24:00:00-06:00:00`, `06:00:00-12:00:00`, `12:00:00-18:00:00`, and `18:00:00-24:00:00`. You can select only one time range.
5. Click **OK**.
   During the selected time range, Apsara Stack Security automatically runs checks based on all check items.

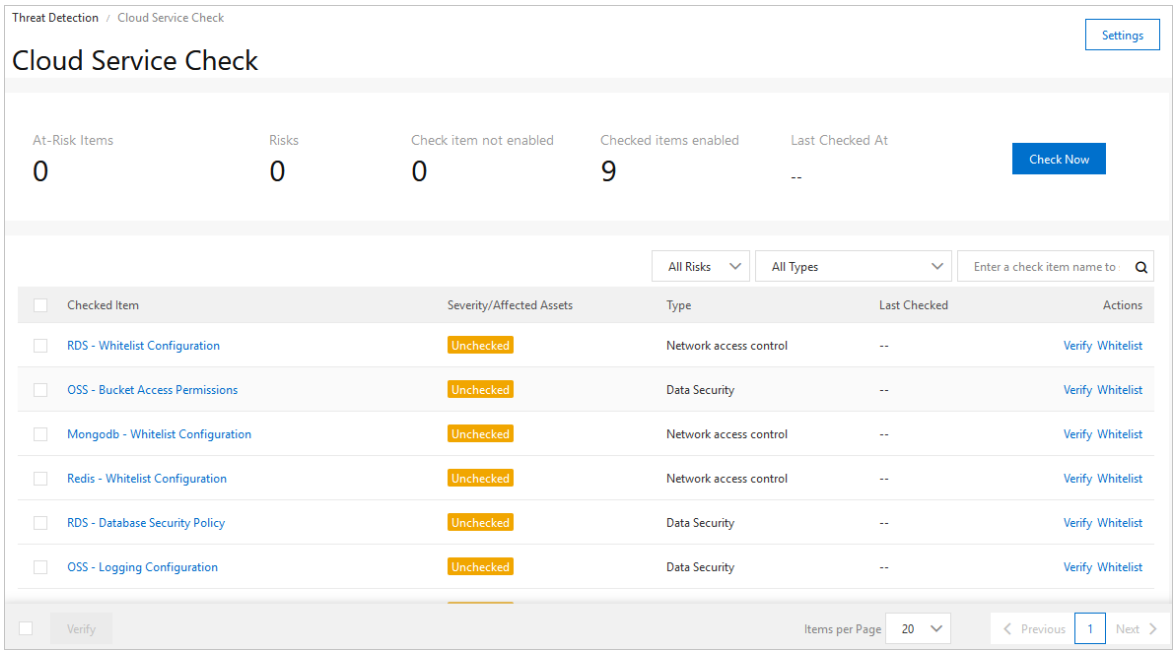
27.1.4.5.3. View and manage check results of Alibaba Cloud services

This topic describes how to view check results and manage configuration risks in Apsara Stack Security Center. Check results include check items, details of check items, potential impacts, and suggestions on how to manage configuration risks. You can manage configuration risks on the Cloud Service Check page.

View check results

1.
2.
3. On the **Cloud Service Check** page, you can view the check results and details of the last configuration

check.



- **View statistics on the result of the last check.**

  Above the check item list, you can view the number of **At-Risk Items** (including the total number of items and the number of items at each risk level), the number of assets with risks (**Risks**), the number of enabled check items, the number of disabled check items, and the last check time.

- **View the check items.**

  In the check item list, you can view information about the check items. The check items include the risk level, the type and number of assets affected by each check item, the type of each check item, and the time of the last check.

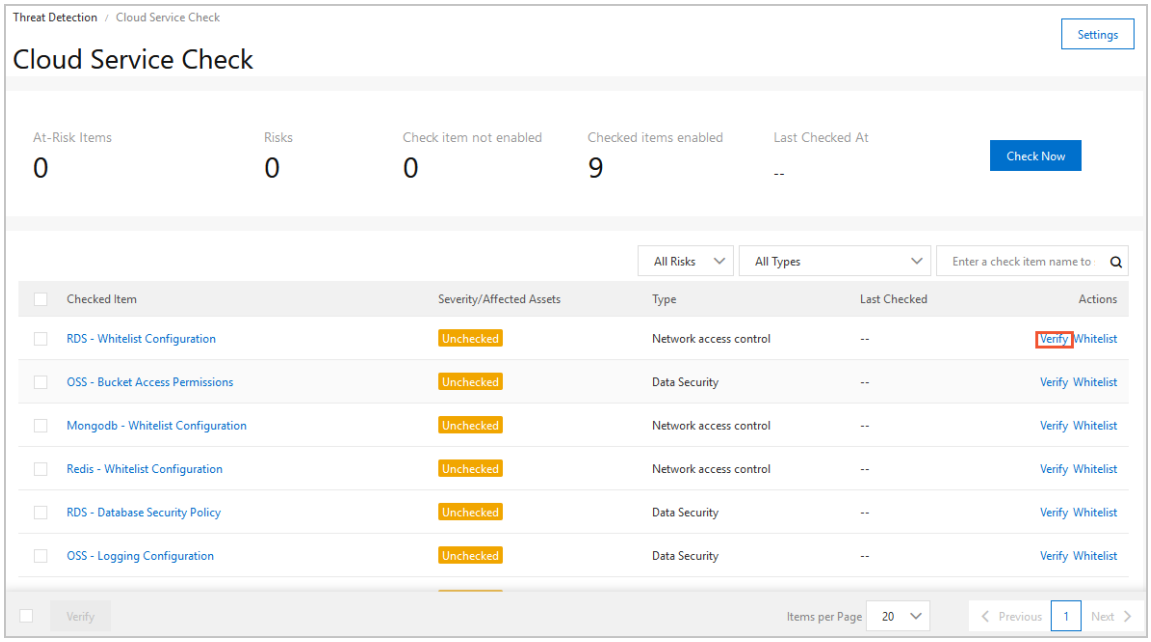
- **View the check result details of a check item.**

  Click the name of the target checked item in the **Checked Item** column to go to the details page. You can view the check description, potential risks, and suggestions on how to manage the risks on this page.

Manage configuration risks

1.
2.
3. On the **Cloud Service Check** page, manage the configuration risks.
   - **Check whether the new settings contain risks.**

If you have changed the configuration settings of an item, find the item on the check item list on the Cloud Service Check page and click **Verify** in the Actions column to check whether the new settings contain risks.



- **Add a check item to the whitelist.**

  If you want to ignore the detected risks of a checked item, find the item on the check item list and click **Whitelist** to add the item to the whitelist. The status of the item will change to **Ignored**. **Ignored** items are not counted as **At-Risk Items**.

  In the check item list, you can also click **Remove** to remove the **Ignored** items from the whitelist.

  > **Note** After you click **Whitelist**, the risk is ignored for this time only. If the risk is detected in the future, Apsara Stack Security Center will continue to report it.

27.1.4.6. Application whitelist

Application whitelists can prevent unauthorized programs from running on your servers and provide a trusted running environment for your assets.

Context

The application whitelist feature allows you to add servers and trusted applications to a whitelist. Applications that are not specified in the whitelist cannot run on your servers. This feature protects your servers from untrusted or malicious programs and improves resource utilization.

After you apply a whitelist policy to a server, Apsara Stack Security Center detects suspicious or malicious processes and generates alerts on the processes that are not specified in the whitelist.

> **Note** An alert is triggered if a process that is not specified in the whitelist is detected. The detected process may be a normal process or a malicious process. If you trust the process that triggers an alert, we recommend that you add the process to the whitelist. A process that is added to the whitelist no longer triggers alerts when it restarts. If the process is malicious, we recommend that you remove this process immediately and check whether configuration files of scheduled tasks have been modified.

Step 1: Create an application whitelist policy

1.

2. In the left-side navigation pane, choose **Threat Detection > Application Whitelists**.
3.
4. On the App Control page, click the **Policies** tab. Then click **Create Policy** in the upper-left corner.
5. In the Create Policy step of the **Create Whitelist Policy** pane, configure the following parameters:
   - **Policy Name**: Enter a whitelist policy name.
   - **Intelligent Learning Duration**: Select a duration for intelligent learning. Valid values: 1 Day, 3 Days, 7 Days, and 15 Days. The intelligent learning feature uses machine learning to automatically collect and categorize large amounts of alert data. Apsara Stack Security Center can identify suspicious or malicious processes based on the collected data.
   - **Servers for Intelligent Learning**: Select the servers that you want to add to the whitelist.
6. Click **Next** to create the whitelist policy.
   After the whitelist policy is created, its details are automatically displayed in the policy list on the Policies tab.

   The following table lists the parameters in the policy list.

Parameter	Description
Policy Name	The name of the created whitelist policy.
Servers	The number of servers to which the whitelist policy is applied.
Status	The status of the policy. Valid values: ○ **Applied**: Intelligent learning is completed. The policy is applied to servers. ○ **Pending Confirmation**: Intelligent learning is completed. The policy must be confirmed and enabled. After intelligent learning is completed, you also need to turn on the switch in the **Policy Status** column to enable this policy. The policy takes effect only after it is enabled. Apsara Stack Security Center automatically identifies the processes on your servers as trusted, suspicious, or malicious processes. ○ **Paused**: Intelligent learning is manually paused. You can click **Continue** in the Actions column to resume intelligent learning. ○ **Progress**: Intelligent learning is in progress. After a whitelist policy is created, Apsara Stack Security Center automatically performs intelligent learning based on the policy. The status of a newly created policy is **Progress**.
Applications	The numbers of processes, including **Trusted**, **Suspicious**, and **Malicious** processes, on all servers to which the policy is applied.

Parameter	Description
Actions	The operations that you can perform on a policy. Valid values: ○ **Apply**: Add or remove servers to which the policy is applied on the **Apply Whitelist Policy** pane. ○ **Modify**: Modify the policy on the **Modify Whitelist Policy** pane. You can change the values of **Policy Name** and **Intelligent Learning Duration**, and modify the servers that need to automatically perform intelligent learning. ○ **Pause Learning**: Pause intelligent learning. ○ **Continue**: Resume intelligent learning. After you click **Continue**, the status of the policy changes to **Progress**. You can view the learning progress of the policy in the **Status** column. ○ **Delete**: Delete the policy. After the policy is deleted, the corresponding servers and processes are no longer protected by the policy.

Step 2: Add servers to the application whitelist

1.
2. In the left-side navigation pane, choose **Threat Detection > Application Whitelists**.
3. On the **Servers** tab of the App Control page, click **Add Server** in the upper-left corner.
4. In the **Add Server** pane, configure the following parameters:
   - **Whitelist Policy**: Select an existing whitelist policy from the drop-down list.
   - **Event Handling**: The default value is **Alert**, which indicates that Apsara Stack Security Center generates alerts when a suspicious process is detected.

     When an unauthorized process starts on a server protected by the whitelist, an alert is automatically triggered. You can click the number in the **Suspicious Events** column to go to the **Alerts** tab and view the alert details.
   - **Servers**: Select the servers that you want to add to the whitelist. You can select multiple servers.

     To search for a server, enter the server name in the search box of **Servers** and click the search icon. Fuzzy match is supported.
5. Click **OK**.
   After you create an application whitelist, you can view the protected servers and the name of the whitelist policy applied to the servers in the server list on the **Servers** tab.

   The following information of the added servers is displayed on the **Servers** tab:
   - **Server Name/IP**: the name and IP address of the server to which the whitelist policy is applied.
   - **Whitelist Policy**: the whitelist policy that is applied to the server.
   - **Suspicious Events**: the number of unauthorized processes that are detected on the server. Apsara Stack Security Center generates alerts immediately when a suspicious process is detected.
   - **Event Handling**: The default value is **Alert**, which indicates that Apsara Stack Security Center generates alerts when a suspicious process is detected.

     When an unauthorized process starts on a server protected by the whitelist, an alert is automatically triggered. You can click the number in the **Suspicious Events** column to go to the **Alerts** tab and view the alert details.
   - **Actions**: You can click **Delete** in the **Actions** column to remove a server from the application whitelist.

     After the server is removed from the whitelist, the application whitelist policy no longer protects the server. Apsara Stack Security Center generates alerts when a process starts on that server.

Add or remove a process to or from an application whitelist

After an application whitelist is configured for your servers, you can view the protected servers and the names of the whitelist policies applied to the servers in the server list on the **Servers** tab. You can click a policy name in the **Whitelist Policy** column to view the processes running on the required server. You can also view the trusted, suspicious, and malicious processes and their detailed information.

The following information about each process on the server is displayed:

- **Type**: the type of the process. Processes are classified as trusted, suspicious, or malicious processes.
- **Process Name**: the name of the process.
- **Hash**: the Hash function of the process. The Hash function is used to ensure that the process is unique and has not been forged.
- **Path**: the file path of the process on the server.
- **Degree of Trustability**: the degree of trustability for the process determined by Apsara Stack Security Center. Valid values: 0% (malicious process), 60% (suspicious process), and 100% (trusted process).

  > **Note** We recommend that you focus on the processes of 0% trustability.

- **Actions**: the operations that can be performed on the process. You can determine whether to add the process to the whitelist based on the services deployed on your server. You can perform the following operations:
  - **Add to Whitelist**: If a process is trusted, add it to the whitelist.
  - **Remove from Whitelist**: After a process is removed from the whitelist, Apsara Stack Security Center identifies the process as untrusted and generates an alert when this process starts.

27.1.4.7. Assets

27.1.4.7.1. View the security status of a server

The Assets page displays security information about each protected server. The information includes the virtual private could (VPC) where each server resides, server status, and risk status. This topic describes how to view the security status of a server. You can specify filter conditions to search for servers and select the items that you want to display on the Assets page.

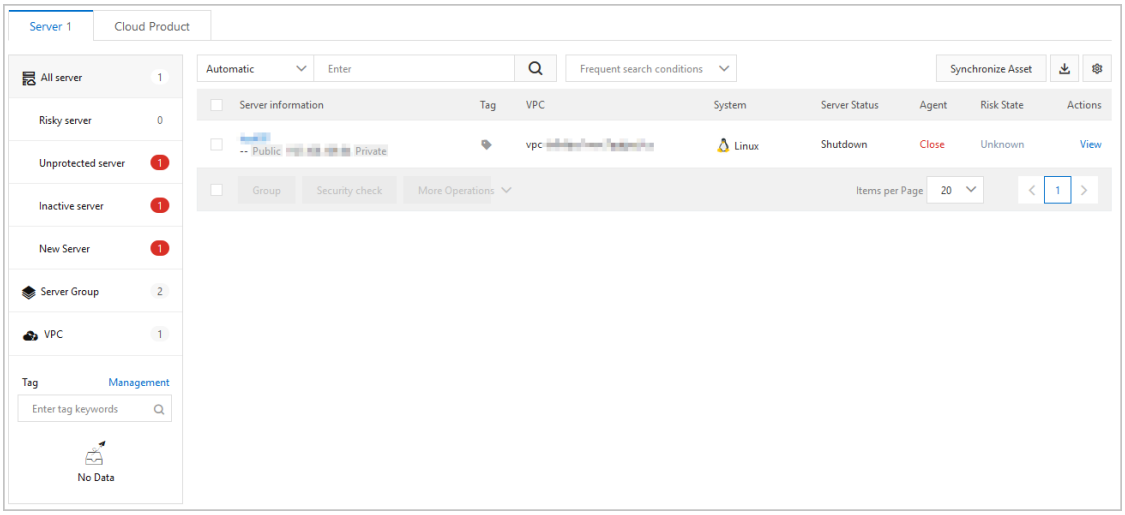
Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. On the **Server** tab, view the security status of each server.

   You can perform the following operations:

   - **Filter servers by status**

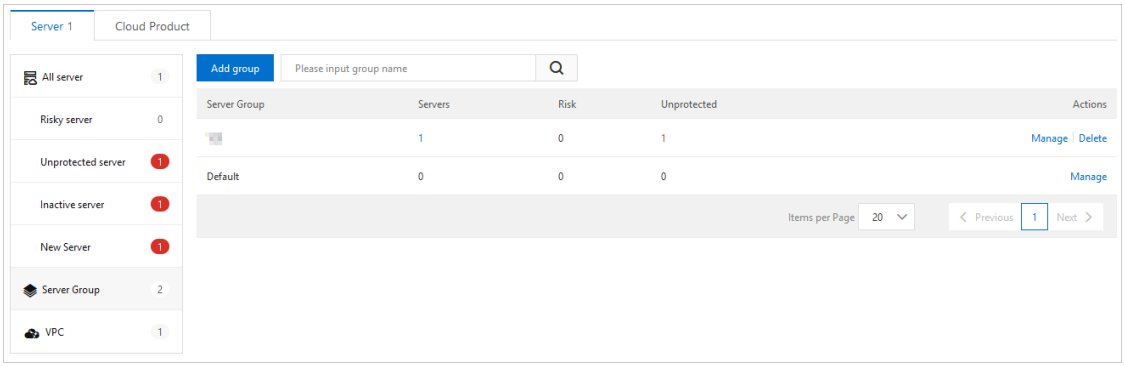
- In **All server**, you can view the numbers of all servers, risky servers, unprotected servers, inactive servers, and new servers.



To view the security information about a server, you can click the name of the server or click **Fix** in the **Actions** column. For more information, see View the details of a single asset.

- You can click **Risky server**, **Unprotected server**, **Inactive server**, or **New Server** to view security information about specific servers.

- **Filter servers by group**
  - You can click **Server Group** to view the numbers of all servers, servers that are at risk, and unprotected servers in each server group. You can also view the total number of server groups.

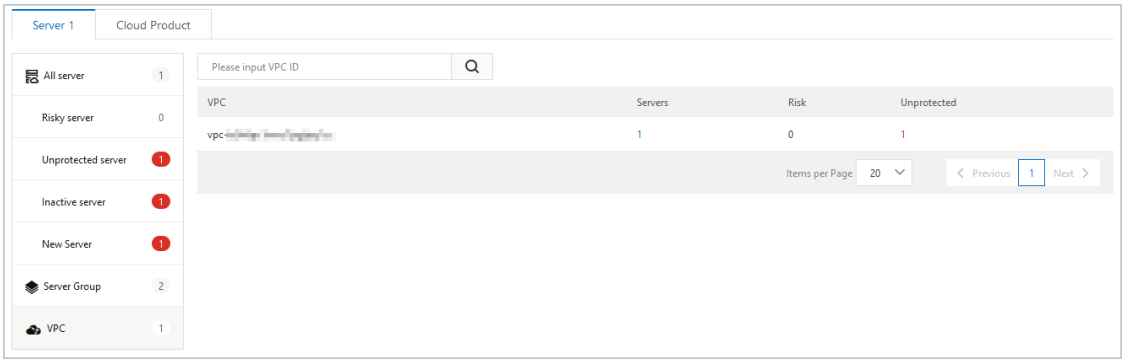


To manage server groups, you can click **Manage** or **Delete** in the Actions column. For more information, see Manage asset groups.

- You can select a server group and click the number in the **Servers**, **Risk**, or **Unprotected** column to view the security information about specific servers in this group.

- **Filter servers by VPC ID**

- You can click **VPC** to view the numbers of all servers, servers that are at risk, and unprotected servers in each VPC. You can also view the total number of VPCs.



- You can select a VPC and click the number in the **Servers**, **Risk**, or **Unprotected** column to view the security information about specific servers in this VPC.

- **Filter servers by tag**

  In the navigation tree, you can click a tag to view the security information about servers to which the tag is added.

- **Filter servers by condition**

  If you click **All server**, **Server Group**, **VPC**, or a tag in the navigation tree, you can specify filter conditions above the right-side list to search for specific servers.

  - Use one filter condition to search for specific servers:

    Above the right-side list, you can select a filter condition and select or enter keywords to search for specific servers. The filter conditions include **Internet IP**, **Private IP**, **Instance name**, **System**, **Baseline problems**, **Vul problems**, **Alert problems**, **Risk Status**, **Online or Offline**, **Tag**, **Group name**, and **OS**.

    > **Note**
    >
    > You can specify multiple filter conditions at a time and specify a Boolean operator for the conditions. The following list describes the Boolean operators:
    >
    > - Boolean operators:
    >   - **AND**: specifies the **AND** logical relation for the conditions.
    >   - **OR**: specifies the **OR** logical relation for the conditions.
    > - If you want to search for servers that meet at least one of the filter conditions, you must set the Boolean operator to **OR**.
    > - If a filter condition requires you to enter a keyword, you must enter the keyword and click the **Search** icon. Results are displayed only after you click the Search icon.

  - Use multiple filter conditions to search for specific servers:

    If you select multiple filter conditions, they are all applied to search for specific servers.

    You can also select **Server Group**, **VPC**, or a tag in the navigation tree and specify filter conditions above the right-side lists to search for specific servers.

- **Save frequently used filter conditions**

  You can save the filter conditions that are applied as frequently used search conditions. To save the conditions, click **Save** above the right-side list, and enter a name in the **Save condition** dialog box. Then, you can select the saved conditions from the **Frequent search conditions** drop-down list to the right of the Search icon.

- **Customize displayed items**

  On the **Assets** page, you can click the icon in the upper-right corner. Then, you can select the items that you want to display on the Assets page.

27.1.4.7.2. View the security status of cloud services

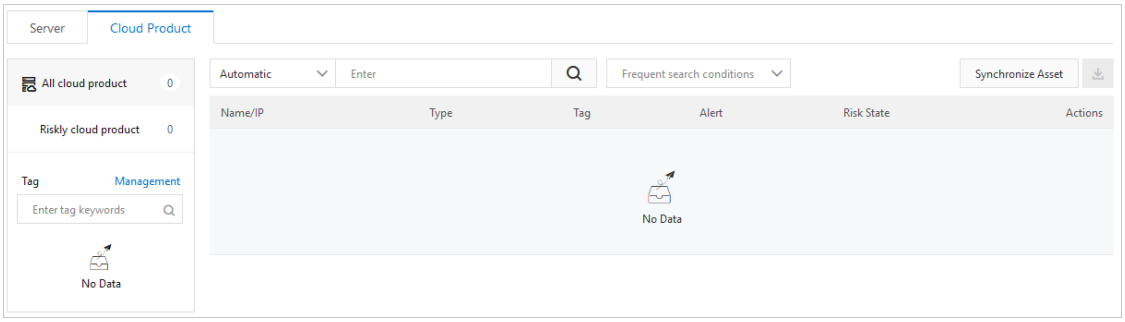
The Assets page displays the security information about each protected cloud service. The information includes the at-risk services and the types of services, for example, SLB and NAT Gateway. This topic describes how to configure search conditions to view the security status of cloud services.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. Click the **Cloud Product** tab to view the security status of cloud services.

   You can perform the following operations based on your business requirements:

   - **Search by asset status**
     - In the left-side pane on the **Cloud Product** tab, you can view the numbers of **All cloud product** and **Risky cloud product**. You can also view the security status of all cloud services.

     - You can click **Risky cloud product** to view the cloud services that are at risk.

     You can click the name of the target cloud service or click **View** in the Actions column that corresponds to a service to view detailed information. For more information, see View the details of a single asset.

   - **Search by asset type**

     Cloud services are classified into two asset types:

     - **SLB**
     - **NAT Gateway**

     In the left-side pane on the Cloud Product tab, you can view the number of cloud services of each asset type. You can click **SLB** or **NAT** to view the security status of the target cloud service.

   - **Search by tag**

     In the **Tag** section in the left-side pane of the Cloud Product tab, you can view the number of assets bound to each tag. You can click a tag on the left of the asset list to view the security status of the cloud services to which this tag is bound.

   - **Filter by search condition**

     You can click **All cloud product**, **SLB**, or **NAT** in the left-side pane of the **Cloud Product** tab and configure search conditions in the search box to search for specific assets.

     For example, you can click **All cloud product** and configure search conditions to search for specific assets.

- Use multiple subconditions to search for specific assets:

  Select a condition from the drop-down list of the search box on the **Cloud Product** tab, and select a subcondition or enter a keyword into the search box to search for specific assets. Supported search conditions are **Internet IP**, **Instance name**, **Alert problems**, **Risk Status**, **Tag**, and **Group name**.

  > Note

- Use multiple search conditions to search for specific assets:

  Apply multiple search conditions.

  - You can click **SLB**, **NAT**, or a tag specified in the **Tag** section and configure conditions in the search box on the **Cloud Product** tab to search for specific assets.
  - You can also click **All cloud product**, **SLB**, or **NAT** and select a tag specified in the **Tag** section to search for specific assets.

- **Set frequently used search conditions**

  You can save applied search conditions as frequently used search conditions. Click **Save** below the search box and enter a name in the **Save condition** dialog box. Then, you can select the saved search condition from the Frequent search conditions drop-down list on the right of the search box.

27.1.4.7.3. View the details of a single asset

The Assets page provides details about all assets. These details include basic information, alert management status, baseline check analysis, and asset fingerprints. This topic describes how to view details of a single server or Alibaba Cloud service.

Context

The basic information about assets is displayed on the **Assets** page. Based on the types of assets, servers or Alibaba Cloud services can be managed in different ways.

The following table lists the features that are supported by servers and Alibaba Cloud services on the **Assets** page. The following content describes the marks that are used to indicate whether a feature is supported by servers or Alibaba Cloud services:

- ×: not supported by this edition.
- √: supported by this edition.

Feature	Description	Server	Cloud service
Basic information	Risk state: displays the number of risks of an asset. Risks can be divided into the following types: • **Vulnerability** • **Alert** • **Baseline risk**	√	√ (Only alerts can be processed.)
	Detail: displays the configuration and protection status of an asset. You can specify a group and a tag for the asset.	√	√ (You cannot specify a group for the asset.)
	Asset investigation: displays asset fingerprints, including ports, software, processes, and accounts.	√	×

Feature	Description	Server	Cloud service
	Vulnerability check: displays the supported types of vulnerability checks. You can enable or disable different types of vulnerability checks for an asset.	√	X
	Logon security setting: displays the common logon locations, IP addresses, time, and accounts of an asset. You can also enable or disable alerts for the asset.	√	X
Vulnerabilities	Displays the vulnerability check results of an asset.	√	X
Alerts	Displays the security alerts of an asset.	√	√
Baseline Risks	Displays the baseline check results of an asset.	√	X
Asset Fingerprints	Displays the fingerprints of an asset.	√	X

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. On the **Assets** page, click the **Server** or the **Cloud Product** tab.
4. On the **Server** or **Cloud Product** tab, find the target asset and click its name.
5. View the details of the target asset.

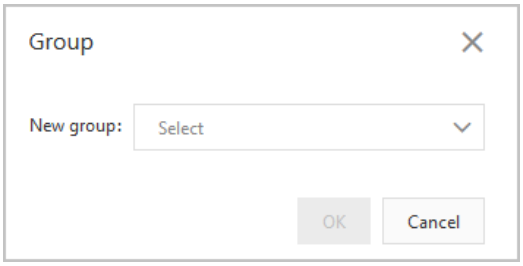
   On the asset details page, click the **Basic Information**, **Vulnerabilities**, **Alerts**, **Baseline Risks**, or **Asset Fingerprints** tab to view relevant details.

   The following content displays the details of the target asset:

   - **Basic Information**: This tab consists of sections where you can view asset details and manage an asset.
     - **Risk State**: This section displays information about vulnerabilities, alerts, and baseline risks of an asset. You can click the number under Vulnerabilities, Alerts, or Baseline Risks to view the details.

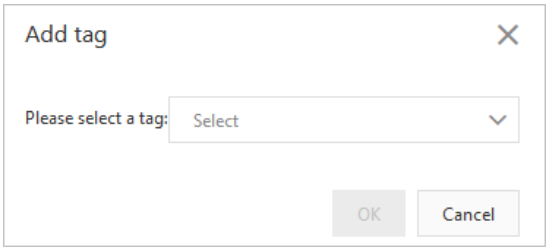
- **Detail**: This section displays information about the asset configuration and security protection settings, and allows you to manage asset tags and groups.
  - **Change asset groups**

    Click **Group**. In the **Group** dialog box, select a new group and click **OK**.

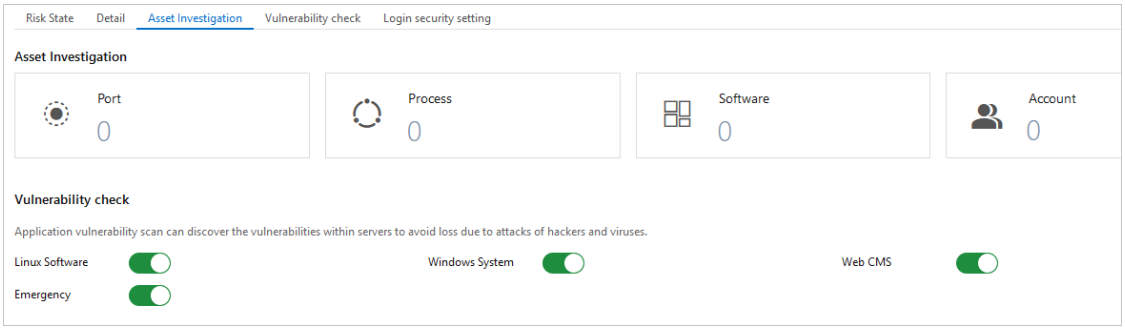
  - **Modify tags**

    Click the icon. In the **Add tag** dialog box, select a tag and click **OK**.

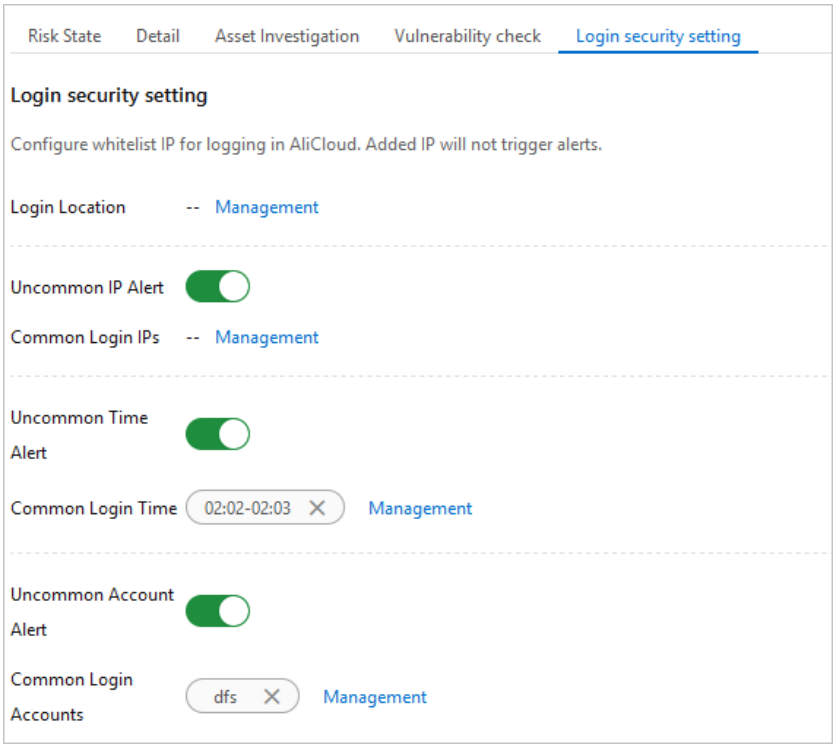
    To remove the tag of an asset, click the icon to the right of the tag.

- **Asset Investigation**: This section displays the fingerprints of an asset. You can click the number under an item to go to the **Asset Fingerprints** tab to view the details.

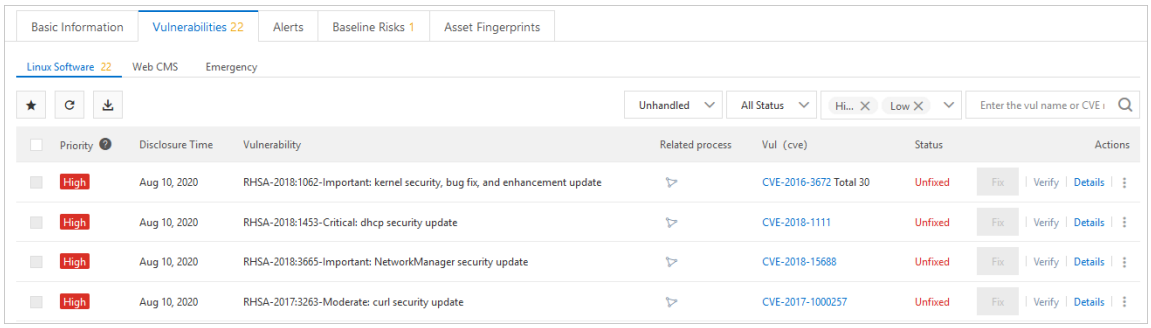
  

- **Vulnerability check**: This section displays vulnerability check items that are enabled or disabled for an asset. You can enable or disable different types of vulnerability checks for the asset. The vulnerabilities include Linux software vulnerabilities, Windows system vulnerabilities, Web CMS vulnerabilities, and emergency vulnerabilities.

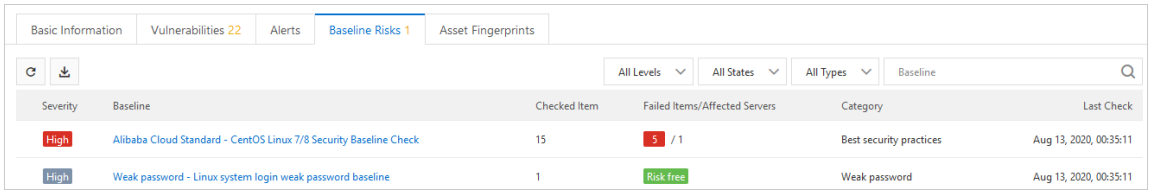
- **Login security setting**: This section allows you to **add approved logon locations**, configure advanced logon settings, **turn on or turn off** the unapproved IP address, time, and account alert function. The advanced logon settings include approved IP addresses or Classless Inter-Domain Routing (CIDR) blocks, time periods, and accounts. You can also add approved IP addresses, time periods, and accounts that are allowed to log on to a specific asset.



- **Vulnerabilities**: This tab displays vulnerabilities of an asset.



- **Alerts**: This tab displays security alerts of an asset.
- **Baseline Risks**: This tab displays baseline risks of an asset.

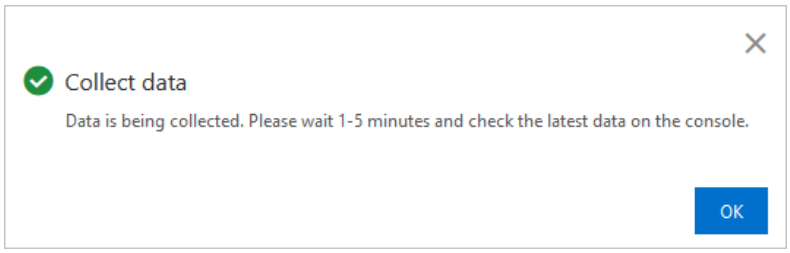


- **Asset Fingerprints**: This tab displays the fingerprints, including ports, processes, software, and accounts of an asset.

  You can manually collect the latest fingerprints of an asset.

  a. You can click the **Port**, **Software**, **Process**, **Account**, or **Scheduled Tasks** tab. In the upper-right corner, click **Collect data now**.

b. In the **Collect data** dialog box, click **OK**.



After the data collection task is submitted, it takes one to five minutes to collect the fingerprints of the target asset. After the data collection task is complete, you can view the latest fingerprints of the target asset.

27.1.4.7.4. Enable and disable server protection

This topic describes how to enable and disable server protection.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. On the **Server** tab, enable or disable server protection for specified servers.
   - **Enable server protection**

     Select one or more servers where the agent is in the **Close** state, and choose **More operations > Turn on protection**.

     After server protection is enabled, the status of the agent on the servers changes to **Enable**.
   - **Disable server protection**

     You can disable server protection for specified servers. Select one or more servers where the **agent** is in the **Enable** state, and choose **More operations > Suspend Protection**.

     > **Note** After server protection is disabled, Apsara Stack Security Center stops providing protection for your servers. The protection mechanisms that are stopped include vulnerability detection and security event alerting. We recommend that you proceed with caution.

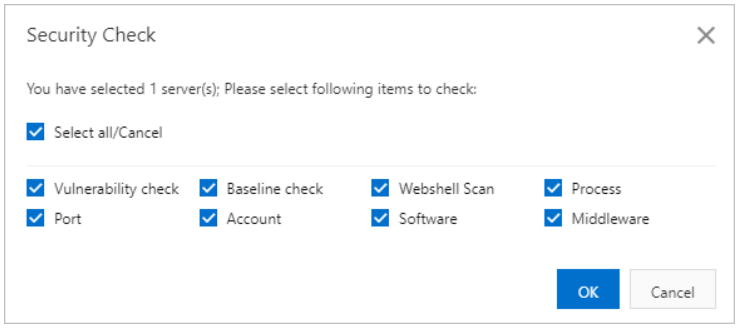
     After server protection is disabled, the status of the **agent** on your servers changes to **Close**.

27.1.4.7.5. Perform a quick security check

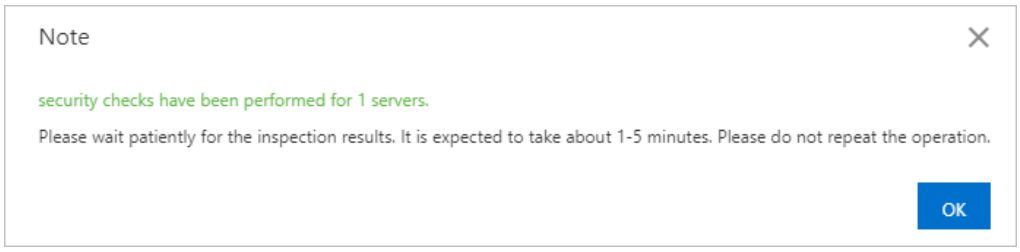
The Server tab of the Assets page allows you to run security checks. You can dispatch security check tasks to scan for vulnerabilities, baseline risks, or webshells, and collect asset fingerprints on a specific server. The asset fingerprints are ports, software, processes, and accounts. This topic describes how to perform a security check on servers.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. On the **Server** tab, select one or more servers on which you want to perform a security check.
4. In the lower part of the page, click **Security check**.
5. In the **Security Check** dialog box, select check items.



6. Click **OK** to start the check.
7. In the message that appears, click **OK**.



After the security check is complete, the check results are automatically displayed on the details pages of the selected servers.

27.1.4.7.6. Manage server groups

This topic describes how to create, modify, delete, and replace server groups.

Create a server group

1.
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. On the **Server** tab, click **Server Group** in the navigation tree.

   > **Note** Ungrouped servers are in the **Default** group.

4. Click **Add group**.
5. In the **Add group** dialog box, configure parameters for the new group.

   To configure the parameters, perform the following steps:

   i. Enter a name for the new group in the **Group name** field.

   ii. Add servers to the new group.

   You can add servers in the Default group to the new group. You can also move servers from another group to the new group. To add or move servers, select **Default** or other groups from the **Select Group** drop-down list, select servers in the groups, and then click the > icon to add the selected servers to the new group.

6. Click **OK**.
   In the server group list, you can view the new group.

Modify or delete a server group

The following procedure describes how to modify or delete a server group. When you modify a server group, you can rename the group or adjust the servers in the group.

1.
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. On the **Server** tab, click **Server Group** in the navigation tree.
4. Find the server group that you want to modify or delete. In the Actions column, click **Manage** or **Delete**.

   You can perform the following operations based on your business requirements:

   - **Modify the group**
     a. In the Actions column, click **Manage**. The Group dialog box appears.
     b. In the **Group** dialog box, select a group from the **Select Group** drop-down list, select servers in the current group on the right, and then click the icon to move the selected servers to the selected group on the left. You can also select servers in the group on the left and click the icon to move the selected servers to the current group on the right.
     c. Click **OK**.
   - **Delete the group**

     In the Actions column, click **Delete**. In the message that appears, click **OK**.

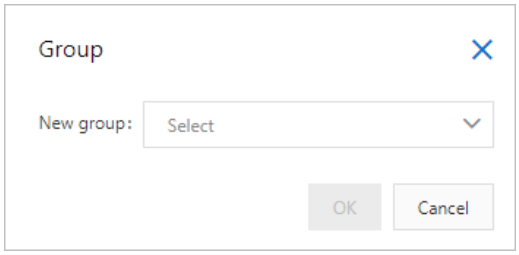
     > **Note** After you delete a group, servers in this group are moved to the **Default** group.

Replace a server group

You can add servers to a server group to manage multiple servers at a time. We recommend that you add the same types of servers to a server group. For example, if you configure a baseline check template, you can specify a server group and apply the template to all servers in the group. You can also filter and view servers based on server groups.

To move servers to a specific server group, perform the following steps:

1.
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. On the Server tab, select one or more servers and click **Group** in the lower part of the page.
4. In the **Group** dialog box, select a new server group.



5. Click **OK**.

27.1.4.7.7. Manage asset tags

This topic describes how to add asset importance tags to your assets and how to create, modify, and delete custom tags.

Context

Apsara Stack Security provides the asset importance tags described in the following table to classify assets. You can select appropriate importance tags for your assets.

An asset importance tag is transformed to an **asset importance score**. An **asset importance score** is used to calculate a vulnerability priority score. You can determine whether to preferentially fix a vulnerability based on the vulnerability priority score. We recommend that you add importance asset tags to core assets. Apsara Stack Security prompts you to fix vulnerabilities based on the importance of each asset. The following table describes the relationships between asset importance tags and asset importance scores.

Asset importance tag	Asset importance score	Recommendation
Important Assets	1.5	Assets that are related to crucial business or store core business data. Virus intrusion into the assets adversely affects the system and causes major loss.
General Assets	1	Assets that are related to non-crucial business and are highly replaceable. Virus intrusion into the assets causes less impact on the system.
Test Assets	0.5	Assets for functional or performance tests, or assets that can cause less impact on the system.

Note If you do not add asset importance tags, the **General Assets** tag is automatically added to each asset. This tag indicates that the asset importance score is 1.

Create a custom tag

1.
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. Click the **Server** or **Cloud Product** tab.
4. In the navigation tree of the **Server** or **Cloud Product** tab, click **Management** to the right of **Tag**.
5. In the **Add tag** dialog box, enter the tag name. In the left-side server list, select servers and click the icon to add the new tag to the selected servers.
6. Click **OK**.

   In the asset list of the Server or Cloud Product tab, you can click the icon in the **Tag** column to add the new tag to an asset.

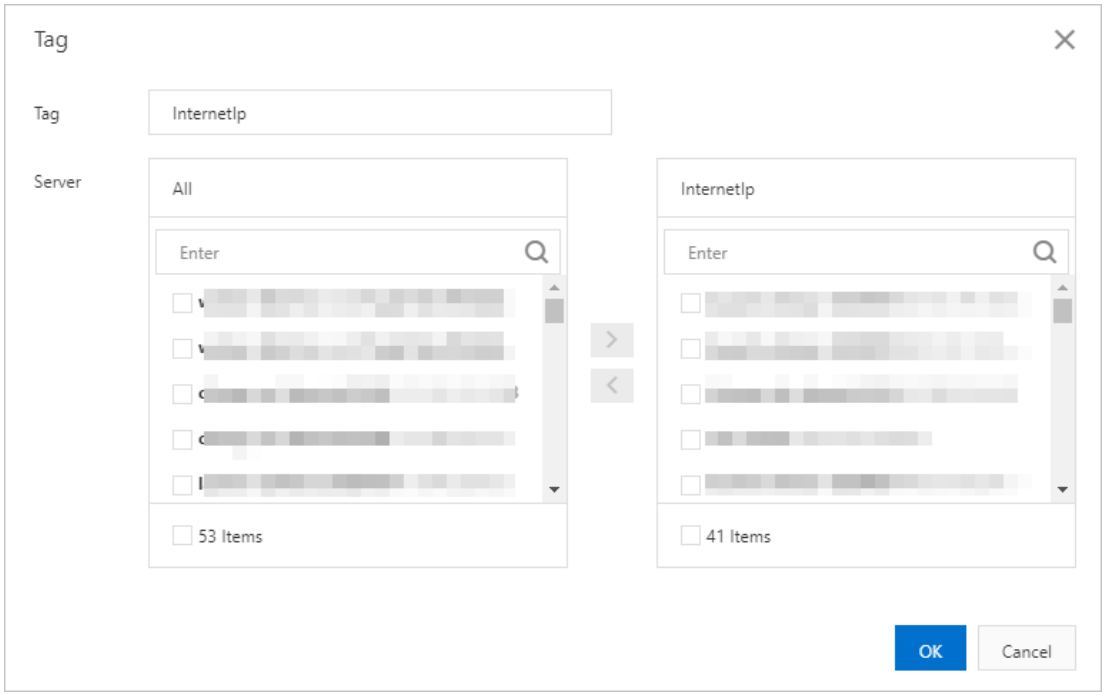
   Note You can add multiple tags to one asset. All tags of an asset are displayed in the **Tag** column.

Modify or delete a custom tag

The following procedure describes how to modify or delete a custom tag. When you modify a tag, you can rename the tag or adjust the servers to which the tag is added.

1.
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. Click the **Server** or **Cloud Product** tab.
4. On the **Server** or **Cloud Product** tab, modify or delete a tag.
   - **Modify a tag**
     a. Find the tag that you want to modify and move the pointer over the icon to the right of the tag.

b. In the **Tag** dialog box, enter a new name in the **Tag** field, add the tag to more servers, or remove the tag from specific servers.



c. Click **OK**.

- **Delete a tag**

  Find the tag that you want to delete and click the × icon in the Tag column. In the message that appears, click **OK**.

27.1.4.8. Vulnerability scan

27.1.4.8.1. Quick start

This topic describes how to get started with the vulnerability scan feature.

The following procedure shows how to use the vulnerability scan feature:

1. Configure the following detection items and the required cycles based on your environment requirements:
   - Overall Monitoring: Configure detection features and the monitoring cycle of each detection feature. For more information, see Configure overall monitoring.
   - Basic Monitoring: Configure Weak Password Vulnerability Monitoring, Operation Security Vulnerability Monitoring, CMS Application Vulnerability Monitoring, and Baseline Monitoring. For more information, see Configure basic monitoring.
   - Web Monitoring: Configure the monitoring cycle and the types of web vulnerabilities that you want to monitor. For more information, see Configure web monitoring.
   - Whitelist: Add the assets that do not need detection to the whitelist. For more information, see Configure a whitelist.
2. Import assets that require vulnerability scans.
   - Import internal assets: Configure a scan engine to import your internal assets in virtual private clouds (VPCs). For more information, see Configure a scan engine for internal assets.
   - Import Internet assets: Directly import Internet assets. For more information, see Import assets.

> **Note** The number of imported assets cannot exceed the specified upper limit.

3. View and confirm the results of vulnerability scans.
   - View the overview information to obtain the general results of vulnerability scans. For more information, see View the information on the Overview page.
   - View and confirm security vulnerability risks. For more information, see .
   - View and confirm host compliance risks. For more information, see Manage security vulnerabilities.
   - View and confirm external risks, such as code leak risks. For more information, see Manage host compliance risks.
4. Generate vulnerability scan reports.

   Generate reports to audit the vulnerabilities and baselines of assets at a regular basis. For more information, see Manage external risks.

27.1.4.8.2. View the information on the Overview page

This topic describes the overall results of vulnerability scans. Security administrators can understand the vulnerability situation based on the overall results.

Context

The vulnerability scan feature can identify the following vulnerabilities: web security vulnerabilities, content management system (CMS) application vulnerabilities, weak password vulnerabilities, operations and maintenance (O&M) security vulnerabilities, and baseline security vulnerabilities.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Overview**.
3. View the overall result of vulnerability scans.

Section	Description
Today's attention	View Asset Overview, Disclosured Risk, and Resolved Risk of the current day. ○ **Asset Overview**: displays the numbers of hosts, websites, and domain names for the current day and provides a security score for the current assets. The radar chart on the right shows the distribution of web security vulnerabilities, CMS application vulnerabilities, weak password vulnerabilities, O&M security vulnerabilities, and baseline security vulnerabilities. ○ **Disclosured Risk**: displays the numbers of high-risk vulnerabilities, medium-risk vulnerabilities, and low-risk vulnerabilities, and the total number of these vulnerabilities for the current day. These vulnerabilities are not fixed. The **Disclosured Risk Distribution** section on the right displays the distribution of unfixed vulnerabilities. ○ **Resolved Risk**: displays the numbers of high-risk vulnerabilities, medium-risk vulnerabilities, and low-risk vulnerabilities, and the total number of these vulnerabilities for the current day. These vulnerabilities are fixed. The **Resolved Risk Distribution** section on the right displays the distribution of fixed vulnerabilities.

Section	Description
Asset Risk Top 5	View the top 5 assets that are at risk on the **Security Vulnerabilities** and **Host Compliance** tabs. These assets are displayed by asset or group.
Risk Monitoring Trend	View the trend charts of vulnerabilities on the **Security Vulnerabilities** and **Host Compliance** tabs. Fixed and unfixed vulnerabilities are identified by lines in different colors. You can move the pointer over a line to view the numbers of unfixed vulnerabilities and fixed vulnerabilities for the specific day.
Asset Monitoring Trend	View the trends in the numbers of protected hosts and websites. Hosts and websites are identified by lines in different colors. You can move the pointer over a line to view the number of protected hosts and websites for the specific day.
Risk Asset Ranking List	View the rankings of assets that are at risk on the **Latest Risk** and **High Risk** tabs.
Port Service Statistics	View the statistics on the **Port** and **Host Service** tabs.

27.1.4.8.3. Asset management

27.1.4.8.3.1. View the results of asset analysis

This topic describes how to view the analysis results of websites and hosts.

Context

The asset analysis feature allows you to view the analysis results of websites and hosts. For the websites, you can view Web Service, Open Source Framework, and Device Type. For the hosts, you can view Host Port, Host Service, and Operation System.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Asset > Asset Analysis**.
3. View websites.
4. View hosts.

27.1.4.8.3.2. Import assets

This topic describes how to import Internet assets.

Context

The vulnerability scan feature works only on imported assets. If you want to scan the vulnerabilities of your assets, you must import your assets.

The assets that the feature supports include Internet assets and internal assets. The internal assets refer to the assets in a virtual private cloud (VPC).

- To import internal assets, you must add a scan engine. For more information, see Configure a scan engine for internal assets.

- To import Internet assets, see this topic.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Asset > Asset Import**.
3. View imported assets.
4. Click **Asset Import** to create an asset import task.
   - To import internal assets, you must add a scan engine. For more information, see Configure a scan engine for internal assets.
   - To import Internet assets, perform the following steps. Elastic IP Address (EIP) assets and Server Load Balancer (SLB) assets are automatically imported, and monitored.
     i. Choose **Internet Assets > Internet Assets**.
     ii. In the **Import Asset** section, select **Manual Import** and enter the required assets in the field. Then, read and select the disclaimer.
     - You can enter domain names, URLs, IP addresses, and CIDR blocks.
     - You can enter multiple assets at a time. Press Enter each time you enter an asset.
     - You cannot enter the assets in VPCs.
     - The number of imported assets must be less than the number of remaining assets supported by the platform.

     > **Note** For example, if the number of remaining assets supported by the platform is 100 and 90 assets are entered, all the assets can be scanned. If 110 assets are entered, only 100 assets can be scanned, and the 10 assets that remain cannot be scanned.

     iii. In the **Asset Info** section, select a group for the imported assets and configure an owner and a tag for the assets.
     - **Asset Group**: Select a group from the drop-down list. You can click the icon to create, edit, or delete a group.
     - **Person in charge**: Select an owner from the drop-down list. You can click the icon to create, edit, or delete an owner.
     - **Asset Tag**: Click **Add Tag** to add a tag to the imported assets.

iv. In the **Import Set** section, select the operations that you want to perform after the assets are imported.

Operation		Description
Asset Discovery	*Auto Import Subdomains*	Automatically queries the subdomain assets of the imported domain names.
	Auto import associated IP	Automatically adds IP addresses that are mapped to the domain names.
	Auto synchronize tags and groups	Applies the group and tag of the imported assets to the assets that are discovered by the system.
Web Asset	*Open WEB Monitoring*	Allows you to enable the web monitoring feature on the imported website assets. If you want to select the web monitoring rules to use, click the icon. In the dialog box that appears, select the required web monitoring rules. For more information about how to configure web monitoring rules, see Configure web monitoring.

v. In the **Whitelist** section, add the assets that do not need to be scanned.

You can enter IP addresses and URLs. If you add more than one asset, you must press Enter each time you enter an asset.

vi. Click **Save**.

5. Manage the new asset import task.

After the asset import task is created, you can view the task in the task list. You can also perform the following operations on the new asset import task.

Icon	Description
	Allows you to view the details, results, and import process of the asset import task.
 or 	Allows you to enable or disable the asset import task.
	Allows you to delete the asset import task.

27.1.4.8.3.3. Manage assets

This topic describes how to view and manage assets.

Context

You can view the information about assets in the asset list. If the purpose or owner of an asset changes, security administrators can move the asset to another group or change the owner.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Asset > Asset List**.
3. View the asset list.

4. Click the **Web Asset** tab and manage websites.
   i. Specify filter conditions to search for specific websites.

      The filter feature allows you to search for the required websites in a more efficient manner.

Filter condition	Description
VPC	The asset type, such as Internet assets or a specific virtual private cloud (VPC).
Asset Source	The source from which the asset is imported. Valid values: **Manual** and **System Find**.
Asset Group	The group to which the asset belongs.
Web Status	The status of the website.
Person in charge	The owner of the website.
Asset Change	The change status of the asset. Valid values: **All**, **New**, **Update**, **No Update**, and **Offline**.
Web Monitoring	The monitoring status of the website.
Risk Level	The risk level of the asset.
Web Service	The service type and version of the website.
WAF Recognition	Specifies whether the asset is identified by WAF.
Open Source Framework	The open source framework type of the asset.
Device Type	The type of the device.
Time Range	The time range during which assets are imported.
Key Information	The crucial information of the asset. The crucial information includes the website, domain name, IP address, title, and tag.

   ii. Click **Export** to export the websites to an Excel file.

   iii. Select multiple websites to manage them.

Action	Description
Batch Web Monitoring	Allows you to enable or disable web monitoring for multiple websites. ■ **Batch Open Monitoring**: To enable web monitoring for multiple websites, select Batch Open Monitoring from the drop-down list of Batch Web Monitoring. ■ **Batch Stop Monitoring**: To disable web monitoring for multiple websites, select Batch Stop Monitoring from the drop-down list of Batch Web Monitoring
Change Group	Allows you to change the asset group of multiple assets at a time.
Change Person in charge	Allows you to change the owner of multiple assets at a time.
Batch Delete	Allows you to delete multiple assets at a time. After assets are deleted, Apsara Stack Security does not scan the assets.

iv. Manage a single website.

Find the website and perform operations allowed in the **Operation** column.

5. Click the **Host Asset** tab and manage the hosts.

i. Specify filter conditions to search for specific hosts.

The filter feature allows you to search for the required hosts in a more efficient manner.

Filter condition	Description
VPC	The asset type, such as Internet assets or a specific VPC.
Asset Source	The source from which the asset is imported. Valid values: **Manual** and **System Find**.
Asset Groups	The group to which the asset belongs.
Person in charge	The owner of the asset.
Asset Change	The change status of the asset. Valid values: **All**, **New**, **Update**, **No Update**, and **Offline**.
Risk Level	The risk level of the asset. Valid values: **All**, **High**, **Middle**, **Low**, and **Security**.
SurviveStatus	The status of the asset. Valid values: **Alive** and **Close**.
Operation System	The operating system of the host.
Host Port	The port of the host.
CDN Recognition	Specifies whether Content Delivery Network (CDN) is configured for the asset.
Host Service	The service of the host.
Time Range	The time range during which assets are imported.
Key Information	The crucial information of the asset. The crucial information includes the IP address, host, tag, and domain name.

ii. Click **Export** to export the hosts to an Excel file.

iii. Select multiple hosts to manage them.

Action	Description
Change Group	Allows you to change the asset group of multiple assets at a time.
Change Person in charge	Allows you to change the owner of multiple assets at a time.
Delete	Allows you to delete multiple assets at a time. After assets are deleted, Apsara Stack Security does not scan the assets.

iv. Manage a single host.

Find the host and perform operations allowed in the **Operation** column.

27.1.4.8.3.4. Manage asset availability

This topic describes how to manage the availability of assets.

Context

If hosts or websites are used for a long period of time, they may become unavailable due to errors. Availability monitoring allows security administrators to discover unavailable assets. Then, the security administrators can troubleshoot the issues that cause the assets to become unavailable.

Availability monitoring supports the following methods:

- HTTP monitoring: This method is used to monitor websites.
- PING monitoring: This method is used to monitor hosts.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Asset > Availability Monitoring**.
3. View availability monitoring tasks.
4. Create an availability monitoring task.

   Availability monitoring supports HTTP monitoring and PING monitoring.

   - To create an HTTP monitoring task, perform the following steps:
     a. Click **Add Monitoring**.
     b. Click the **HTTP Monitoring** tab.
     c. Configure the following parameters.

Parameter	Description
Monitoring Name	The name of the availability monitoring task.
Monitoring Target	The website that you want to monitor.
Monitoring Frequency	The interval at which you want to monitor the website. Valid values: **1 Minute**, **5 Minute**, **15 Minute**, and **30 Minute**.
Request Method	The request method that is used to send HTTP request packets. Valid values: **HEAD**, **GET**, **POST**, and **PUT**.
Alert Setting	The policy based on which Apsara Stack Security reports alerts. If one of the following conditions is met, the website is unavailable: ■ Response Time: If the actual response time is greater than the specified value, an exception occurs. ■ Response Status: If an unexpected status code is returned, an exception occurs.

     d. Click **Save**.
   - To create a PING monitoring task, perform the following steps:
     a. Click **Add Monitoring**.
     b. Click the **PING Monitoring** tab.

c. Configure the following parameters.

Parameter	Description
Monitoring Name	The name of the availability monitoring task.
Monitoring Target	The host that you want to monitor.
Monitoring Frequency	The interval at which you want to monitor the host. Valid values: **1 Minute**, **5 Minute**, **15 Minute**, and **30 Minute**.
Alert Setting	The policy based on which Apsara Stack Security reports alerts. If one of the following conditions is met, the host is unavailable: ■ Response Time: If the actual response time is greater than the specified value, an exception occurs. ■ Response Status: If an unexpected status code is returned, an exception occurs.

d. Click **Save**.

5. Manage multiple availability monitoring tasks at a time.

   You can manage multiple availability monitoring tasks in the monitoring task list at a time.

   - Start multiple availability monitoring tasks at a time

     Select multiple availability monitoring tasks and choose **Batch Monitoring Manage > Batch Open Monitoring**.

   - Stop multiple availability monitoring tasks at a time

     Select multiple availability monitoring tasks and choose **Batch Monitoring Manage > Batch Stop Monitoring**.

   - Delete multiple availability monitoring tasks at a time

     Select multiple availability monitoring tasks and choose **Batch Monitoring Manage > Batch Delete Monitoring**.

27.1.4.8.3.5. Manage custom update detection tasks

This topic describes how to manage custom update detection tasks.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Asset > Custom Update Detection**.
3. View custom update detection tasks.
4. Create a custom update detection task.
   i. Click **Add Detection**.
   ii. On the **Add Custom Update Detection** page, configure the following parameters.
   iii. Click **Save**.

27.1.4.8.4. Risk management

27.1.4.8.4.1. Manage security vulnerabilities

This topic describes how to view and handle security vulnerabilities detected by the vulnerability scan feature.

Context

On the **Security Vulnerability** page, security administrators can view the security vulnerabilities detected by the vulnerability scan feature.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Risk > Security Vulnerability**.
3. View security vulnerabilities.
   - Click the **Disclosured** or **Resolved** tab to view **unfixed vulnerabilities** or **fixed vulnerabilities**.
   - View risk statistics. The statistics cover **Unfixed Risks**, **Recovered Risks**, **Unconfirmed Risks**, **Confirmed Risks**, and **Ignored Risks**.
4. Specify search conditions to view specific security vulnerabilities. The conditions include the **VPC** and **Risk Level** parameters.
5. Handle security vulnerabilities.

   Security administrators can analyze and confirm whether security vulnerabilities affect the security of assets based on the vulnerability information.
   - Confirm risks

     If a vulnerability affects the security of assets, confirm the risk after the security vulnerability is fixed.

     a. Find the vulnerability and click the  icon in the **Actions** column.

     b. In the drop-down list, select **Confirm Risk**.

     c. In the dialog box that appears, click **OK**.
   - Ignore risks

     If a vulnerability is a false positive or does not affect the security of assets, ignore the risk.

     a. Find the vulnerability and click the  icon in the **Actions** column.

     b. In the drop-down list, select **Ignore Risk**.

     c. In the dialog box that appears, click **OK**.
6. Click **Export** to export the security vulnerabilities to your local computer.

27.1.4.8.4.2. Manage host compliance risks

This topic describes how to view and confirm host compliance risks.

Context

On the **Host Compliance** tab, security administrators can view the host compliance issues that are detected by the vulnerability scan feature.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Risk**. On the page that appears, click the **Host Compliance** tab.
3. View host compliance risks.

You can click the **Disclosured** and **Resolved** tabs to view **unfixed risks** and **fixed risks**.

4. Specify search conditions to search for host compliance risks. The conditions include the **VPC** and **Risk Level** parameters.
5. Handle host compliance risks.

   Security administrators can analyze and confirm whether host compliance risks affect the security of assets based on the risk information.

   - Confirm risks

     If a host compliance risk affects the security of assets, harden the security of hosts and confirm the risk.

     a. Find the vulnerability and click the icon in the **Operation** column.

     b. In the drop-down list, select **Confirm Risk**.

     c. In the dialog box that appears, click **OK**.

   - Ignore risks

     If a host compliance risk proves to be a false positive or does not affect the security of assets, ignore the risk.

     a. Find the vulnerability and click the icon in the **Operation** column.

     b. In the drop-down list, select **Ignore Risk**.

     c. In the dialog box that appears, click **OK**.
6. Click **Export** to export the host compliance risks to your local computer.

27.1.4.8.4.3. Manage external risks

This topic describes how to view and identify external risks, such as code leaks.

Prerequisites

A GitHub account is available, and the tokens of the account are obtained.

Context

On the **External Risk** page, security administrators can use the vulnerability scan feature to check whether the GitHub library has risks of code leaks.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Risk > External Risk**.
3. Click the **Code Disclosure** tab.
4. Add the assets that you want to monitor.

   i. In the **Asset Monitoring List** section, click **Token setting**.

   ii. In the **Enter GitHub Token** dialog box, enter the tokens.

   You can enter multiple tokens. Press Enter each time you enter a token.

   iii. Click **Add Asset** and select the external assets that you want to monitor.
5. View the assets that have risks of code leaks.

   You can view the unfixed and fixed risks on the **Resolved** and **Disclosured** tabs.
6. Handle the risks of code leaks.

   Security administrators can analyze and check whether the security of the assets is affected based on the risk

information.

- Confirm risks

  If the risks of code leaks affect the security of your assets, harden the hosts and confirm the risks.

- Ignore risks

  If the risks of code leaks prove to be false positives or do not affect the security of your assets, ignore the risks.

27.1.4.8.4.4. Create a custom risk detection task

This topic describes how to create a custom risk detection task.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Risk > Custom Risk Detection**.
3. Click **Add Detection**.
4. On the **Add Custom Risk Detection** page, configure the following parameters.

Parameter	Description
Detection Name	The name of the custom risk detection task.
Detection Target	The asset on which you want to perform risk detection. Valid values: **Private network asset** and **Public network asset**.
Emergency Detection	The switch that is used to enable or disable the emergency detection feature. If you enable this feature, you can select emergency detection items from the detection item list.
Basic Risk Detection	The switch that is used to enable or disable the basic risk detection feature. For more information about how to configure this feature, see Configure basic monitoring.
WEB Risk Detection	The switch that is used to enable or disable the web risk detection feature. For more information about how to configure this feature, seeConfigure web monitoring.

5. Click **Save**.

27.1.4.8.5. Report management

27.1.4.8.5.1. Create a report

This topic describes how to create a report.

Context

A security administrator can create a report to view the security statuses of specific assets during a period of time and implement security measures as required.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Report**.
3. Click **Risk Report**.
4. Click **Add Report**.
5. Configure the following parameters.
6. Click **Create**.

Result

After the report is created, it appears in the report list on the **Report** page.

27.1.4.8.5.2. Delete multiple reports at a time

This topic describes how to delete multiple reports at a time.

Context

You can delete multiple reports that are no longer required at a time to save storage space.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Report**.
3. Click **Risk Report**.
4. In the report list, select the reports that you want to delete.
5. Click **Batch Delete**.

27.1.4.8.6. Configuration management

27.1.4.8.6.1. Configure overall monitoring

This topic describes how to configure overall monitoring for the vulnerability scan feature. Overall monitoring includes Asset Monitoring Configuration, Base Risk Monitoring Configuration, External Risk Monitoring Configuration, and Scan Configuration.

Context

Overall monitoring allows you to configure detection features and the monitoring cycle for each detection feature.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Configuration > Monitoring Configuration > Overall Monitoring**.
3. In the **Monitoring Status** section, view the status of overall monitoring.
4. Configure detection features.

   Detection features include **Asset Monitoring Configuration**, **Base Risk Monitoring Configuration**, **External Risk Monitoring Configuration**, and **Scan Configuration**.

   In this step, the **Asset Monitoring Configuration** detection feature is used as an example.

i. Turn on Asset Monitoring Configuration to enable the asset monitoring feature.
   - After the switch is turned on, the switch is in the **On** state. When in the On state, the switch is blue. After the switch is turned off, the switch is in the **Off** state. When in the Off state, the switch is gray.
   - You must turn on **Asset Monitoring Configuration** and **Base Risk Monitoring Configuration** to enable the two features. External Risk Monitoring Configuration and Scan Configuration are automatically enabled.

ii. Configure the following parameters.

Asset Monitoring Configuration

Parameter	Description
Monitoring Item	The item that you want to monitor. Valid value: **Subdomain Discovery**. If you want to import assets, you can set the Import Set parameter to **Auto Import subdomains**. Then, subdomains are automatically imported. If you select **Subdomain Discovery**, Apsara Stack Security regularly discovers subdomains for assets whose Import Set parameter is set to **Auto Import subdomains**.
Monitoring Cycle	The cycle based on which you want to perform detection. Valid values: **customization**, **per week**, and **per month**. ■ **customization**: Specify the interval at which you want to perform detection. Unit: days. ■ **per week**: Specify the days of each week on which you want to perform detection. ■ **per month**: Specify the days of each month on which you want to perform detection.
Detection Time	The time when you want to perform detection. The time varies based on the value of the **Monitoring Cycle** parameter. ■ If you set the **Monitoring Cycle** parameter to **customization**, select a time range of the day in which you want to perform detection. ■ If you set the **Monitoring Cycle** parameter to **per week**, select the days of each week and the time range in which you want to perform detection. ■ If you set the **Monitoring Cycle** parameter to **per month**, select the days of each month and the time range in which you want to perform detection. **Note** For example, if you set the **Monitoring Cycle** parameter to **per week** and select **Monday** to **Sunday** and 00:00:00 to 24:00:00 for the **Detection Time** parameter, Apsara Stack Security performs detection 24 hours a day, 7 days a week.
Port Range	The ports on which you want to perform detection. Valid values: **customization**, **Full port**, **TOP100**, and **TOP1000**. ■ **customization**: Specify the ports to scan. ■ **Full port**: Scan all ports. ■ **TOP100**: Scan top 100 ports. You can click **Add** to add more ports. ■ **TOP1000**: Scan top 1,000 ports. You can click **Add** to add more ports.

Parameter	Description
Host Alive Detection Settings	The option that is used to check whether a host is running. By default, the ping feature is used to check whether a host is running. If the host has the ping feature disabled, the status of the host is detected based on top 20 ports and custom ports. To specify custom ports, click **Settings**. In the Host Alive Detection Settings dialog box, specify the ports in the **Custom Port** field.

Base Risk Monitoring Configuration

Parameter	Description
Monitoring Item	The item that you want to monitor. Valid values: **Weak Password**, **Common Vulnerabilities**, **Baseline Monitoring**, and **Host Compliance**. ■ **Weak Password**: Attackers can guess passwords or launch brute-force attacks to crack passwords. Then, the attackers can obtain relevant permissions. If you select this item, weak password vulnerabilities can be identified. ■ **Common Vulnerabilities**: Web security vulnerabilities and CMS application vulnerabilities are included. If you select this item, common vulnerabilities can be identified. Then, you can install patches at the earliest opportunity. ■ **Baseline Monitoring**: Risks in host configuration and account configuration are detected. ■ **Host Compliance**: Host compliance risks are detected.
Monitoring Cycle	The cycle based on which you want to perform detection. Valid values: **customization**, **per week**, and **per month**. ■ **customization**: Specify the interval at which you want to perform detection. Unit: days. ■ **per week**: Specify the days of each week on which you want to perform detection. ■ **per month**: Specify the days of each month on which you want to perform detection.
Detection Time	The time when you want to perform detection. The time varies based on the value of the **Monitoring Cycle** parameter. ■ If you set the **Monitoring Cycle** parameter to **customization**, select a time range of the day in which you want to perform detection. ■ If you set the **Monitoring Cycle** parameter to **per week**, select the days of each week and the time range in which you want to perform detection. ■ If you set the **Monitoring Cycle** parameter to **per month**, select the days of each month and the time range in which you want to perform detection. **Note** For example, if you set the **Monitoring Cycle** parameter to **per week** and select **Monday** to **Sunday** and 00:00:00 to 24:00:00 for the **Detection Time** parameter, Apsara Stack Security performs detection 24 hours a day, 7 days a week.

External Risk Monitoring Configuration

Parameter	Description
Monitoring Item	The item that you want to monitor. Valid value: **Code Disclosure**. If you select Code Disclosure, Apsara Stack Security detects leaked source code of your assets.
Monitoring Cycle	The cycle based on which you want to perform detection. Valid values: **customization**, **per week**, and **per month**. ■ **customization**: Specify the interval at which you want to perform detection. Unit: days. ■ **per week**: Specify the days of each week on which you want to perform detection. ■ **per month**: Specify the days of each month on which you want to perform detection.
Detection Time	The time when you want to perform detection. The time varies based on the value of the **Monitoring Cycle** parameter. ■ If you set the **Monitoring Cycle** parameter to **customization**, select a time range of the day in which you want to perform detection. ■ If you set the **Monitoring Cycle** parameter to **per week**, select the days of each week and the time range in which you want to perform detection. ■ If you set the **Monitoring Cycle** parameter to **per month**, select the days of each month and the time range in which you want to perform detection. Note For example, if you set the **Monitoring Cycle** parameter to **per week** and select **Monday** to **Sunday** and 00:00:00 to 24:00:00 for the **Detection Time** parameter, Apsara Stack Security performs detection 24 hours a day, 7 days a week.

Scan Configuration

Parameter	Description
Risk Re-detection	The time at which you want to perform detection again. If risks are detected in specific assets, Apsara Stack Security scans the assets again each day at the time you specify.
Asset Scanning Rate	The scan rate. Valid values: **Slow Mode**, **General Mode**, **Fast Mode**, and **Turbo Mode**.
Risk Scanning Rate	The scan rate. Valid values: **Slow Mode**, **General Mode**, **Fast Mode**, and **Turbo Mode**.
UserAgent Setting	The User-Agent property.

iii. Click **Save**.

27.1.4.8.6.2. Configure basic monitoring

This topic describes how to configure basic monitoring.

Context

Basic monitoring includes Weak Password Vulnerability Monitoring, Operation Security Vulnerability Monitoring, CMS Application Vulnerability Monitoring, and Baseline Monitoring.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Configuration**. On the page that appears, click the **Monitoring Configuration** tab. Then, click the **Basic Monitoring** tab.
3. Click **Weak Password Vulnerability Monitoring** and configure rules to monitor weak password vulnerabilities.
   - By default, all monitoring items use the default weak password library.
   - To disable a monitoring item, perform the following step:

     Find the monitoring item that you want to disable and click the icon in the **Operation** column.
   - To enable a monitoring item, perform the following step:

     Find the monitoring item that you want to enable and click the icon in the **Operation** column.
   - To customize weak passwords for a monitoring item, perform the following steps. In this example, **MySQL Weak Password Vulnerability** is used.
     a. In the **Default Weak Password** column, turn off the switch. The switch status changes to .
     b. In the **Operation** column, click the icon.
     c. In the **Customize MySQL Weak Password** dialog box, customize weak passwords.
     d. Click **Yes**.
   - To apply the same custom weak passwords to multiple monitoring items, perform the following steps:
     a. In the **Default Weak Password** column, turn off the switches for the monitoring items that you want to apply the same custom weak passwords. The switch status changes to .

     > **Note** If you want to apply a custom weak password to a monitoring item, you must turn off the switch in the Default Weak Password column of the monitoring item.

     b. Click **Tailored Overall Weak Password**.
     c. In the **Tailored Overall Weak Password** dialog box, customize weak passwords.
     d. Click **Yes**.
4. Click **Operation Security Vulnerability Monitoring** and configure operations and maintenance (O&M) security vulnerability monitoring.

No.	Description
1	The switch that is used to enable or disable the **Operation Security Vulnerability Monitoring** feature. We recommend that you enable this feature to improve system security.
2	The switch that is used to enable or disable a monitoring item. You can disable monitoring items based on your business requirements.

5. Click **CMS Application Vulnerability Monitoring** and configure content management system (CMS) application vulnerability monitoring.

No.	Description
1	The switch that is used to enable or disable the **CMS Application Vulnerability Monitoring** feature. We recommend that you enable this feature to improve system security.
2	The switch that is used to enable or disable a monitoring item. You can disable monitoring items based on your business requirements.

6. Click **Baseline Monitoring** and configure baseline monitoring.

   To add a baseline monitoring item, perform the following steps:

   i. Click **Add**.

   ii. In the **Add Baseline** dialog box, configure the baseline monitoring item.

   In this example, a baseline monitoring item is added to block Telnet-based access.

Parameter	Description
Baseline Name	The name of the baseline monitoring item. Example: Block Telnet-based access.
Baseline Rule	The detection rule that is used by the baseline monitoring item. This rule checks whether hosts use disabled ports or run disabled services. Valid values: ■ **Port Disabled**: ports that you want to disable. Example, 23. ■ **Service Disabled**: services that you want to disable. Example: Telnet.
Baseline Range	The scope of assets to which the baseline monitoring item can be applied. Valid values: **Private IP** and **NatIP**. You must configure this parameter and select specific assets.

   iii. Click **Yes**.

27.1.4.8.6.3. Configure web monitoring

This topic describes how to configure web monitoring.

Context

Web monitoring allows you to configure monitoring items for monitoring web vulnerabilities. You can also configure conditions to block website crawlers.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Configuration**. On the page that appears, click the **Monitoring Configuration** tab and then the **Web Monitoring** tab.
3. View existing rules.

   > **Note** **Default rules** are created by the system. You can only view details of the default rules, but cannot modify or delete them.

4. Create a web monitoring rule.

   i. Click **Add Rule**.

ii. On the **Add Web Monitoring Rule** page, configure the following parameters.

Parameter	Description
Rule Name	The name of the web monitoring rule.
Monitoring Cycle	The monitoring cycle. Valid values: **Customization**, **Per Week**, and **Per Month**. ■ **Customization**: Specify the interval at which you want to perform detection. ■ **Per Week**: Specify the days of each week on which you want to perform detection. ■ **Per Month**: Specify the days of each month on which you want to perform detection.
Detection Time	The detection time varies based on the value of the **Monitoring Cycle** parameter. ■ If you set the **Monitoring Cycle** parameter to **Customization**, select the time range of the day in which you want to perform detection. ■ If you set the **Monitoring Cycle** parameter to **Per Week**, select the days of each week and the time range in which you want to perform detection. ■ If you set the **Monitoring Cycle** parameter to **Per Month**, select the days of each month and the time range in which you want to perform detection. **Note** For example, if you set the **Monitoring Cycle** parameter to **Per Week** and select **Monday** to **Sunday** and 00:00:00 to 24:00:00 for the **Detection Time** parameter, Apsara Stack Security performs detection 24 hours a day, 7 days a week.
Monitoring Options	The type of web vulnerabilities that you want to monitor. Supported operations: **Select All**, **Inverse**, and **Clear**.
UserAgent	The User-Agent field of the HTTP request packet. The User-Agent field identifies the application type, operating system, software developer, and version number of the proxy software that initiates requests.
Cookies	The cookie parameters.
Key Page	The web directories or pages that you want to monitor.
Excluded Page	The web directories or pages that you do not want to monitor.
Crawler Depth	The capturing depth of crawlers. Valid values: **10**, **15**, and **30**.
URL Numbers	The number of URLs that are used for crawling. Valid values: **500**, **1000**, and **2000**.
Scanning Frequency	The scan frequency of web monitoring. Valid values: **Request 10 Times Per Second** and **Request 15 Times Per Second**.

iii. Click **Yes**.

5. Manage the web monitoring rule.

Icon	Description
	Modify the rule.
	Delete the rule.
Batch Delete	If you want to delete multiple rules, select the rules you want to delete and click **Batch Delete**.

27.1.4.8.6.4. Configure a whitelist

This topic describes how to configure a whitelist.

Context

Apsara Stack Security does not scan the assets that are added to a whitelist. Before you add assets to a whitelist, make sure that the assets are secure.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Configuration**. On the page that appears, click the **Monitoring Configuration** tab. Then, click the **Whitelist** tab.
3. View the whitelists.

   > **Note** By default, the whitelist feature is enabled. If you do not want to use the whitelist feature, turn off the switch in the upper-right corner.

4. Create a whitelist.
   i. Click **Add**.
   ii. In the **Add Whitelist** dialog box, configure the following parameters.
   - If you select **Asset Group** for the **Whitelist** parameter, select a group from the second drop-down list. The assets in this group are added to the whitelist.
   - If you select **Customization** for the **Whitelist** parameter, enter the IP addresses or URLs that you want to add to the whitelist in the field that appears.

   iii. Click **Yes**.
5. Manage whitelists.
   - Delete a whitelist

     Find a whitelist and click the  icon in the **Operation** column.

   - Delete multiple whitelists at a time

     Select multiple whitelists and click **Batch Delete**.

27.1.4.8.6.5. Configure a scan engine for internal assets

This topic describes how to configure a scan engine for internal assets, such as the assets of a virtual private cloud (VPC).

Context

You must add a scan engine for a VPC before you can scan for vulnerabilities on the assets of the VPC.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Vulnerability Scan > Configuration**. On the page that appears, click the **Scan Engine Manage** tab. Then, click the **Private-sector assets** tab.
3. Click the name of the VPC whose assets you want to scan.
4. Click **Add Scan Engine**.
5. In the **Add Scan Engine** dialog box, select a vSwitch for the VPC from the **vSwitch** drop-down list.
6. Click **OK**.

27.1.4.9. Create a security report

Security reports help monitor the security status of your assets. You can specify the content, types of statistics, and email addresses of recipients to create a security report. This topic describes how to create a security report.

Procedure

1.
2. In the left-side navigation pane, choose **Threat Detection > Security Reports**.
3. On the **Reports** page, click **Create Report**.

   > **Notice** In addition to the default security report created by Apsara Stack Security, you can create a maximum of nine security reports.

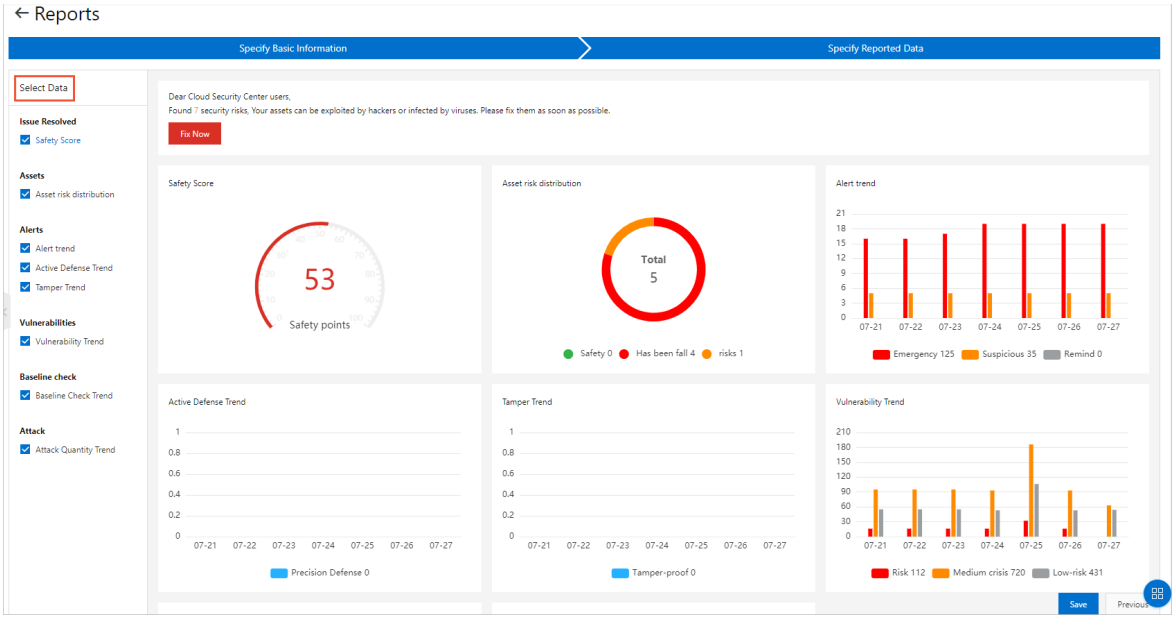
4. In the **Specify Basic Information** step, configure the parameters.

   Configure the following parameters:

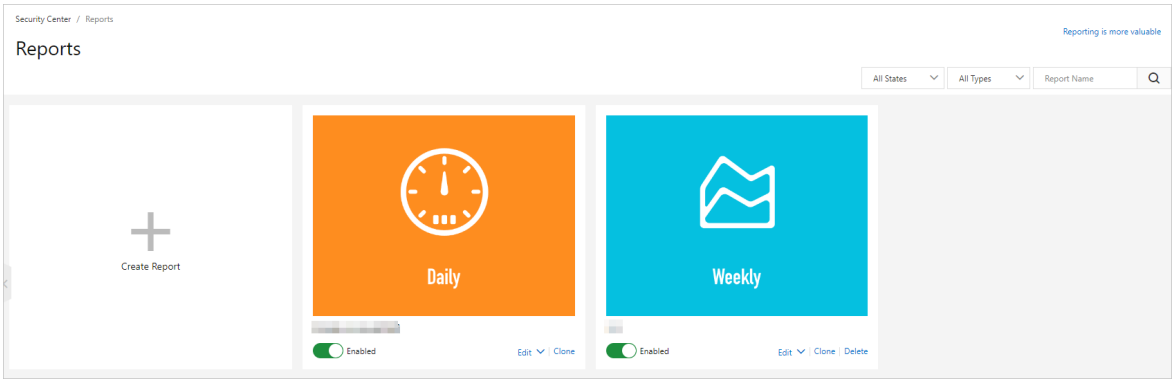
   - **Report Name**: Enter a name for the security report.
   - **Report Type**: Select a report type from the drop-down list. Valid values: *Daily*, *Weekly*, *Monthly*, and *Custom*.

     If you select *Custom*, you must also set the **Data Collection Period** parameter to specify the cycle on which data is collected.

5. Click **Next**.
6. In the **Specify Reported Data** step, select the types of data that you want to view in the security report. You can select assets, alerts, vulnerabilities, baselines, attacks, and other data related to security operations.



7. Click **Save**.
   You can view the newly created security report on the **Reports** page.



27.1.5. Network Traffic Monitoring System

27.1.5.1. View traffic trends

This topic describes how to view the network traffic trends, inbound traffic statistics, and outbound traffic statistics.

Context

By analyzing traffic trends, the security administrator can obtain the throughput and the peaks and troughs of traffic periods. In addition, the security administrator can block traffic from malicious IP addresses by viewing the top five IP addresses with the most traffic.

Procedure

1.
2. In the left-side navigation pane, choose **Network Security > Traffic Analysis > Traffic Trend**.
3. In the upper-right corner of the **Traffic Trends** page, select the time range, which can be **Last 1 Hour**, **Last 24 Hours**, or **Last 7 Days**.
4. View network traffic information.
   - Network traffic trends

View the network traffic trends from the selected time range. The network traffic trends include inbound and outbound traffic measured in bit/s.

- Inbound Traffic

  View the information of Inbound Sessions, Inbound Applications, and Destination IPs with Most Requests.

- Outbound Traffic

  View the information of Outbound Sessions, Outbound Applications, and Source IPs with Most Requests.

5. (Optional)Click the icon to export traffic trends as a PDF file.

27.1.5.2. View traffic at the Internet border

This topic describes how to view traffic at the Internet border. You can obtain up-to-date information about network security.

Prerequisites

The Network Traffic Monitoring System module is purchased and deployed at the egress (ISW) of Apsara Stack. **This module audits, analyzes, and manages both inbound and outbound traffic at Internet borders.**

Context

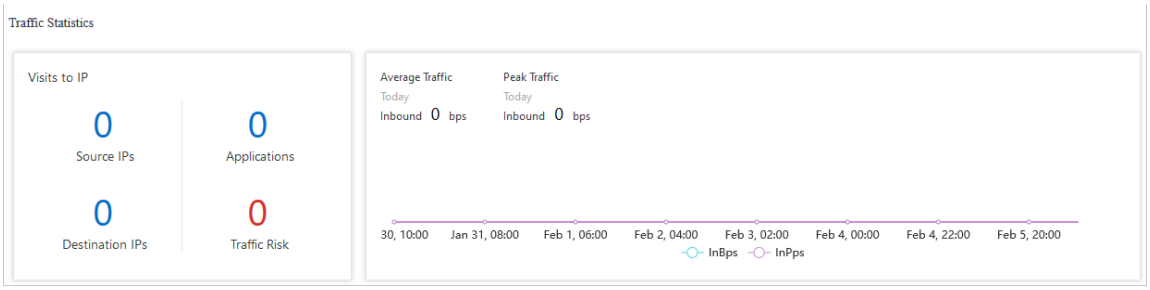
You can use traffic information to identify abnormal Internet traffic and block malicious traffic.

Procedure

1.
2. In the left-side navigation pane, choose **Network Security > Traffic Analysis > Internet Border**.
3. Specify traffic filter conditions.

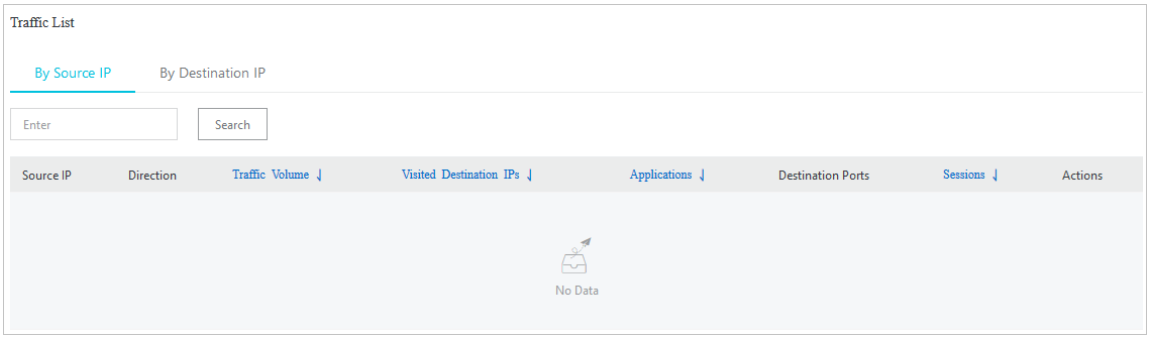
Item	Description
1	Specify the traffic direction. Valid values: *Inbound* and *Outbound*. ○ *Inbound*: The traffic flows from the Internet to the internal network. ○ *Outbound*: The traffic flows from the internal network to the Internet.
2	Specify whether you want to view the traffic from the IP address or application dimension. Valid values: *By IP* and *By Application*.
3	Specify the time range. Valid values: *Last 1 Hour*, *Last 24 Hours*, and *Last 7 Days*.

4. View details about the traffic at the Internet border.
   - **Traffic Statistics**



- The **Visits to IP** section includes **Source IPs**, **Destination IPs**, **Applications**, and **Traffic Risk**.
- In the traffic chart on the right, you can view **Average Traffic**, **Peak Traffic**, and traffic trends.

- **Traffic List**



In the **Traffic List** section, you can view traffic details.

5. In the **Traffic List** section, view abnormal traffic of the specified IP address.
   - If *Inbound* is specified, you can view abnormal traffic on the **By Destination IP** tab of the **Traffic List** section.
   - If *Outbound* is specified, you can view abnormal traffic in the **Traffic List** section.

27.1.5.3. View traffic at the internal network border

This topic describes how to view the traffic at the internal network border. You can obtain up-to-date information about network security based on the traffic.

Prerequisites

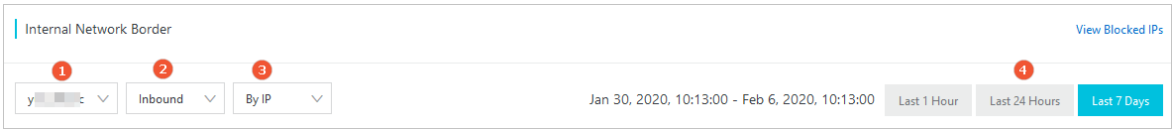
The Network Traffic Monitoring System module is purchased and deployed at the ingress (CSW) of Apsara Stack. **This module is used to audit, analyze, and control both inbound and outbound traffic routed over leased lines between on-premises data centers and virtual private clouds (VPCs).**

Context

You can use traffic information to identify suspicious traffic from the internal network and block malicious requests.

Procedure

1.
2. In the left-side navigation pane, choose **Network Security > Traffic Analysis > Internal Network Border**.
3. Specify traffic filter conditions.



Item	Description
1	Select a VPC name from the drop-down list.
2	Specify the traffic direction. Valid values: *Inbound* and *Outbound*. ○ *Inbound*: The traffic flows from the Internet to the internal network. ○ *Outbound*: The traffic flows from the internal network to the Internet.
3	Specify whether you want to view the traffic that flows through the internal network border from the IP address or application dimensions. Valid values: *By IP* and *By Application*.
4	Specify the time range. Valid values: *Last 1 Hour*, *Last 24 Hours*, and *Last 7 Days*.

4. View details about the traffic at the internal network border.
   - **Traffic Statistics**
     - The **Visits to IP** section includes **Source IPs**, **Destination IPs**, **Applications**, and **Traffic Risk**.
     - In the traffic chart on the right, you can view **Average Traffic**, **Peak Traffic**, and traffic trends.
   - **Traffic List**

     In the **Traffic List** section, you can view traffic details.

     If **By IP** is specified, you can view the suspicious traffic of the specified IP address in the **Traffic List** section.

27.1.5.4. Create packet capture tasks

This topic describes how to create a packet capture task. You can enable the packet capture feature to capture network data packets for specific IP addresses and ports and analyze the packets. This way, you can locate faults, analyze attacks, and identify security risks to network communications.

Procedure

1.
2. In the left-side navigation pane, choose **Network Security > Traffic Analysis > Packet Capture**.
3. On the **Packet Capture** page, click **Create Packet Capture Task**.
4. In the **Create Packet Capture Task** panel, configure parameters and click **OK**.

Parameter	Description
Task Name	The name of the packet capture task. We recommend that you enter an informative name, such as a name that indicates the purpose of the task.
Maximum Bytes	The maximum number of bytes in a packet that can be captured. If the number of bytes in a packet exceeds this value, the excessive bytes are discarded.
Duration (s)	The maximum duration for the packet capture task. Unit: seconds.

Parameter	Description
Protocol	The transmission protocol of packets. Valid values: ○ **All** ○ **TCP** ○ **UDP** ○ **ICMP**
IP Address Type	The IP protocol of packets. Valid values: IPV4 and IPV6.
Direction	The direction of packets. Valid values: **Bidirectional**, **In**, and **Out**.
IP	The IP address of packets.
Port	The port of packets.

Result

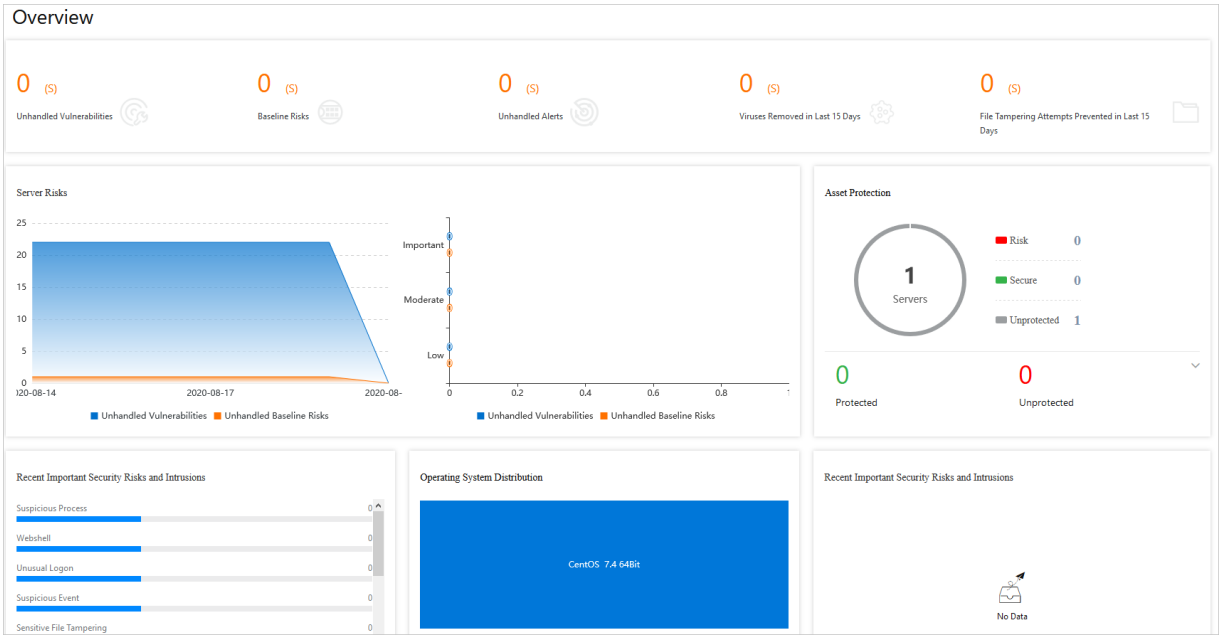
You can go to the **Packet Capture** page to view the newly created packet capture task and the task status.

27.1.6. Server security

27.1.6.1. Server security overview

The security administrator can view the current security status of all servers on the server security overview page of Apsara Stack Security Center.

In the left-side navigation pane, choose **Server Security > Overview**. On the page that appears, you can view detailed information in the Overview, Server Risks, Asset Protection, Operating System Distribution, and Recent Important Security Risks and Intrusions sections.



- **Overview**: This section displays the number of security vulnerabilities of each type (including **Unhandled Vulnerabilities** and **Baseline Risks**) and the number of security events of each type (including **Unhandled Alerts**, **Viruses Removed in Last 15 Days**, and **File Tampering Attempts Prevented in Last 15 Days**) on servers.

- **Server Risks**: This section displays the number of unhandled vulnerabilities, the number of baseline risks, and the distribution of risk levels.
- **Asset Protection**: This section displays the number of protected servers and the number of offline servers.
- **Recent Important Security Risks and Intrusions**: This section displays the recent important risks and events on your servers. You can click a risk or event to view the details.

27.1.6.2. Server fingerprints

27.1.6.2.1. Manage listener ports

This topic describes how to regularly collect information from listener ports on a server, and record and view the port changes and historical port information. This allows you to locate suspicious listening behavior.

Context

This task is suitable to the following scenarios:

- Check for servers that listen on a specific port.
- Check for ports that are open on a specific server.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Servers > Server Fingerprints**. On the page that appears, click the **Port** tab.
3. View **Port**, **Protocol**, and **Server**.

   You can search for a port by using its port number, process name, server name, or IP address.
4. Click a port number to view the details, such as the assets and protocol.

27.1.6.2.2. Manage software versions

This topic describes how to regularly collect software version information of a server and record the changes. This helps to check your software assets.

Context

This task is suitable to the following scenarios:

- Check for software assets that are installed without authorization.
- Check for software of outdated versions.
- Locate the affected assets when vulnerabilities are detected.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Servers > Server Fingerprints**. On the page that appears, click the **Software** tab.
3. View all software in use and the number of servers that use such software.

   You can search for a piece of software by using its software name, version, installation directory, server name, or IP address.
4. Click a software name to view the details, such as the assets and software version.

27.1.6.2.3. Manage processes

This topic describes how to regularly collect the process information on a server and record changes. This helps check for processes and view historical process changes.

Context

This task is suitable to the following scenarios:

- Checks for servers that run a specified process.
- Checks for processes that run on a server.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Servers > Server Fingerprints**. On the page that appears, click the **Process** tab.
3. View all running processes and the number of servers that run these processes.

   You can search for a process by using its **process name**, **running user**, **startup parameter**, or **server name or IP address**.
4. Click a process name to view the details, such as the assets, path, and startup parameters.

27.1.6.2.4. Manage account information

This topic describes how to regularly collect the account information on a server and record changes. This helps check for accounts and view historical account changes.

Context

This task is suitable to the following scenarios:

- Check for servers where the specified account is created.
- Check for accounts that are created on a server.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Servers > Server Fingerprints**. On the page that appears, click the **Account** tab.
3. View all logged-on accounts and the number of servers that use these accounts.

   You can search for an account by using its account name, root permissions, server name, or IP address.
4. Click an account name to view the details, such as the assets, root permissions, and user group.

27.1.6.2.5. Manage scheduled tasks

This topic describes how to regularly collect information of scheduled tasks on a server. This allows you to check your tasks.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Servers > Server Fingerprints**. On the page that appears, click the **Scheduled Tasks** tab.
3. View the paths of all tasks and the number of servers that run these tasks.

   You can search for a task by using its path, server name, or IP address.
4. Click a task path to view the details, such as the assets, executed command, and task cycle.

27.1.6.2.6. Set the server fingerprint collection frequency

You can set the frequency at which the data of running processes, system accounts, listening ports, and software versions is collected.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Servers > Server Fingerprints**. In the upper-right corner of the page that appears, click **Settings**.
3. Select the collection frequency from each drop-down list.
4. Click **OK** to complete the configuration.

27.1.6.3. Threat protection

27.1.6.3.1. Vulnerability management

27.1.6.3.1.1. Manage Linux software vulnerabilities

This topic describes how to manage Linux software vulnerabilities.

Context

Apsara Stack Security automatically scans the software that are installed on your servers based on the vulnerabilities provided in the Common Vulnerabilities and Exposures (CVE) list. It also sends you alerts about the detected vulnerabilities. In addition, Apsara Stack Security provides commands that are used to fix vulnerabilities and allows you to verify these vulnerability fixes.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Vulnerabilities**. On the page that appears, click the **Linux Software** tab.
3. View the detected Linux vulnerabilities.

   > **Note** You can locate a vulnerability by using the search and filter functions.

4. Click a vulnerability to go to the vulnerability details page. You can view details about the vulnerability and affected assets on this page.

   > **Note** You can locate specific affected assets by using the search and filter functions.

   - **Basic Information**: the basic information of the vulnerability, including the name, Common Vulnerability Scoring System (CVSS) score, description, and resolution.
   - **Affected Assets**: the servers that are affected by the vulnerability.
5. Select an action based on the impact of the vulnerability.

   Actions on vulnerabilities

Option	Description
Generate Fix Command	Select this option to generate the commands that are used fix the vulnerability. You can then log on to the server to run these commands.

Option	Description
Fix Now	Select this option to fix the vulnerability.
Restarted and Verified	If a vulnerability fix takes effect only after a server reboot, you must reboot the server after the status of the vulnerability changes to **Fixed (To Be Restarted)**. After the reboot, click **Restarted and Verified**.
Ignore	Select this option to ignore a vulnerability. The system does not send you an alert about an ignored vulnerability.
Verify	Click **Verify** to verify the vulnerability fix. If you do not manually verify a fix, the system automatically verifies the fix within 48 hours after the vulnerability fix is complete.

You can fix a vulnerability for one or more affected assets at a time.

- To fix a vulnerability for one affected asset, select an action from the **Actions** column of the asset.
- To fix a vulnerability for one or more affected assets, select the target servers, and select an action in the lower-left corner.

27.1.6.3.1.2. Manage Windows vulnerabilities

This topic describes how to manage Window vulnerabilities.

Context

Apsara Stack Security automatically checks if your servers have the latest Microsoft updates installed, and notifies you of the detected vulnerabilities. Apsara Stack Security also automatically detects and fixes major vulnerabilities on your servers.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Vulnerabilities**. On the page that appears, click the **Window System** tab.
3. Check the detected Windows vulnerabilities.

   > **Note** You can find a vulnerability by using the search and filter functions.

4. Click a vulnerability to go to the vulnerability details page. You can view details about the vulnerability and affected assets on this page.

   > **Note** You can find specific affected assets by using the search and filter functions.

   - **Basic Information**: the basic information of the vulnerability, including the name, Common Vulnerability Scoring System (CVSS) score, description, and resolution.
   - **Affected Assets**: the servers that are affected by the vulnerability.
5. Select an action based on the impact of the vulnerability. Actions on vulnerabilities describes the actions.

   Actions on vulnerabilities

Option	Description

Option	Description
Rectify	Select this option to fix the vulnerability. The system caches an official Windows patch in the cloud for your server to download and update.
Ignore	Select this option to ignore a vulnerability. The system does not send you an alert about an ignored vulnerability.
Verify	Click **Verify** to verify the vulnerability fix.
Restarted and Verified	If a vulnerability fix takes effect only after a server reboot, you must reboot the server after the status of the vulnerability changes to **Fixed (To Be Restarted)**. After the reboot, click **Restarted and Verified**.

You can fix a vulnerability for one or more affected assets at a time.

- To fix a vulnerability for one affected asset, select an action from the **Actions** column of the asset.
- To fix a vulnerability for one or more affected assets, select the target servers, and select an action in the lower-left corner.

27.1.6.3.1.3. Manage Web CMS vulnerabilities

This topic describes how to manage Web CMS vulnerabilities.

Context

The Web CMS vulnerability detection feature obtains the information of the latest vulnerabilities and provides patches in the cloud. This helps you detect and fix vulnerabilities.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Vulnerabilities**. On the page that appears, click the **Web CMS** tab.
3. View all vulnerabilities.

   > **Note** You can find a vulnerability by using the search and filter functions.

4. Click a vulnerability to go to the vulnerability details page. You can view details about the vulnerability and affected assets on this page.

   > **Note** You can find specific affected assets by using the search and filter functions.

5. Select an action based on the impact of the vulnerability. Actions on vulnerabilities describes the actions.

   Actions on vulnerabilities

Option	Description
Rectify	Select this option to fix the Web CMS vulnerability by replacing the web files that contain the vulnerability on your server. **Note** Before you fix the vulnerability, we recommend that you back up the web files affected by this vulnerability. For more information about the paths of the web files, see the paths specified in the vulnerability remarks.

Option	Description
Ignore	Select this option to ignore a vulnerability. The system does not send you an alert about an ignored vulnerability.
Verify	Click **Verify** to verify the vulnerability fix. If you do not manually verify a fix, the system automatically verifies the fix within 48 hours after the vulnerability fix is complete.
Undo Fix	For vulnerabilities that have been fixed, click **Undo Fix** to restore the web files that have been replaced.

You can fix a vulnerability for one or more affected assets at a time.

- To fix a vulnerability for one affected asset, select an action from the **Actions** column of the asset.
- To fix a vulnerability for one or more affected assets, select the target servers, and select an action in the lower-left corner.

27.1.6.3.1.4. Manage emergency vulnerabilities

This topic describes how to manage emergency vulnerabilities.

Context

Apsara Stack Security automatically detects vulnerabilities on servers, such as the unauthorized Redis access vulnerability and Struts S2-052 vulnerability, and sends vulnerability alerts. After you fix a vulnerability, you can also check whether the fix is successful.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Vulnerabilities**. On the page that appears, click the **Emergency** tab.
3. View all vulnerabilities.

   You can quickly locate a vulnerability by using the search and filter functions.
4. Click a vulnerability to go to the vulnerability details page. You can view detailed vulnerability information and affected assets on this page.

   You can quickly locate specific affected assets by using the search and filter functions.
5. Select an action based on the impact of the vulnerability. Actions on vulnerabilities describes the actions.

   Follow the instructions to manually fix the vulnerabilities on the **Emergency** tab.

   Actions on vulnerabilities

Action	Description
Ignore	Ignore a vulnerability. The system does not alert you about an ignored vulnerability.
Verify	**Verify** the fix after you manually fix a vulnerability. If you do not manually verify a fix, the system automatically verifies the fix within 48 hours after the vulnerability fix is complete.

You can fix a vulnerability for one or more affected assets at a time.

- To fix a vulnerability for one affected asset, select an action from the **Actions** column of the asset.

- To fix a vulnerability for one or more affected assets, select the target servers, and select an action in the lower-left corner.

27.1.6.3.1.5. Configure vulnerability management policies

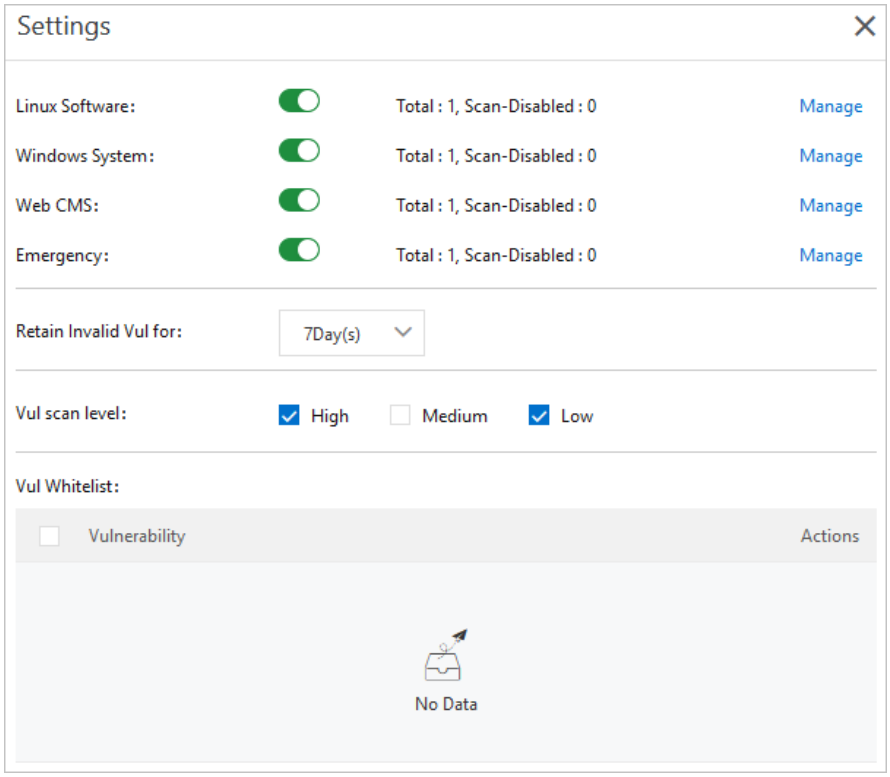
You can enable or disable automatic detection for different types of vulnerabilities, and enable vulnerability detection for specific servers. You can also set a time duration for which invalid vulnerabilities are retained, and configure a vulnerability whitelist.

Context

A vulnerability whitelist allows you to exclude vulnerabilities from the detection list. You can add multiple vulnerabilities in the vulnerability list to the whitelist. The system does not detect vulnerabilities in the whitelist. You can manage the vulnerability whitelist on the vulnerability settings page.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Vulnerabilities**.
3. In the upper-right corner, click **Settings** to configure vulnerability management policies.



- Select a vulnerability type and enable or disable detection for vulnerabilities of this type.
- Click **Manage** next to a vulnerability type and specify the servers on which vulnerabilities of this type are detected.
- Select a time duration for which invalid vulnerabilities are retained: 7 days, 30 days, or 90 days.

  > **Note** If you do not take any action on a detected vulnerability, the system determines that the alert is invalid. The system deletes the vulnerability after the specified duration.

- Select the vulnerability severities for scanning.
  - **High**: Vulnerabilities of this severity must be fixed as soon as possible.

- **Medium**: Vulnerabilities of this severity can be fixed later.
- **Low**: Vulnerabilities of this severity do not need to be fixed for now.

- Select vulnerabilities in the whitelist and click **Remove** to enable the system to detect these vulnerabilities and send alerts again.

27.1.6.3.2. Baseline check

27.1.6.3.2.1. Baseline check overview

The baseline check feature automatically checks the security configurations on servers and provides the detailed check results and suggestions for baseline reinforcement.

Description

After you enable the baseline check feature, Apsara Stack Security automatically checks for risks related to the operating systems, accounts, databases, passwords, and security compliance configurations of your servers, and provides reinforcement suggestions. For more information, see Baseline check items.

By default, a full baseline check is automatically performed from 00:00 to 06:00 every day. You can create and manage scan policies for baseline checks. When you create or modify a policy, you can customize the check items, interval, and time period of a baseline check, and select the servers to which you want to apply this policy. For more information, see Add a custom baseline check policy.

Precautions

The following check items are disabled by default. To check these items, make sure that these items do not affect your business and select them when you customize a scan policy.

- Check items related to weak passwords for specific applications such as MySQL, PostgreSQL, and SQL Server

  Note If these check items are enabled, the system attempts to log on to servers with weak passwords. The logon attempts consume server resources and generate many logon failure records.

- Check items related to China classified protection of cybersecurity
- Check items related to the Center for Internet Security (CIS) standard

Baseline check items

Category	Check item
Database	Alibaba Cloud Standard - MongoDB Security Baseline Check
	Alibaba Cloud Standard - Redis Security Baseline Check
	Alibaba Cloud Standard - Oracle 11g Security Baseline Check
	Alibaba Cloud Standard - Memcached Security Baseline Check
	Alibaba Cloud Standard - Mysql Security Baseline Check

Category	Check item
Operating system	Security baseline check against the Alibaba Cloud standard: • Alibaba Cloud Aliyun Linux 2 Benchmark • Alibaba Cloud Standard - CentOS Linux 6 Security Baseline Check • Alibaba Cloud Standard - CentOS Linux 7/8 Security Baseline Check • Alibaba Cloud Standard - Debian Linux 8 Security Baseline • Alibaba Cloud Standard - Red Hat Enterprise Linux 6 Security Baseline Check • Alibaba Cloud Standard - Red Hat Enterprise Linux 7 Security Baseline Check • Alibaba Cloud Standard - Ubuntu Security Baseline • Alibaba Cloud Standard - Windows Server 2008 R2 Security Baseline Check • Alibaba Cloud Standard - Windows 2012 R2 Security Baseline • Alibaba Cloud Standard - Windows 2016/2019 R2 Security Baseline
	Security baseline check against the CIS standard: • Alibaba Cloud Aliyun Linux 2 CIS Benchmark • CIS CentOS Linux 6 LTS Benchmark • CIS CentOS Linux 7 LTS Benchmark • CIS Debian Linux 8 Benchmark • CIS Ubuntu Linux 14 LTS Benchmark • CIS Ubuntu Linux 16/18 LTS Benchmark • CIS Microsoft Windows Server 2008 R2 Benchmark • CIS Microsoft Windows Server 2012 R2 Benchmark • CIS Microsoft Windows Server 2016/2019 R2 Benchmark
	Baseline check on compliance of China classified protection of cybersecurity level II: • Aliyun Linux 2 Baseline for China classified protection of cybersecurity-Level II • CentOS Linux 6 Baseline for China classified protection of cybersecurity-Level II • CentOS Linux 7 Baseline for China classified protection of cybersecurity-Level II • Debian Linux 8 Baseline for China classified protection of cybersecurity-Level II • Redhat Linux 7 Baseline for China classified protection of cybersecurity-Level II • Ubuntu 14 Baseline for China classified protection of cybersecurity-Level II • Linux Ubuntu 16/18 Baseline for China classified protection of cybersecurity-Level II • Windows 2008 R2 Baseline for China classified protection of cybersecurity-Level II • Windows 2012 R2 Baseline for China classified protection of cybersecurity-Level II • Windows 2016/2019 R2 Baseline for China classified protection of cybersecurity-Level II

Category	Check item
	Baseline check on compliance of China classified protection of cybersecurity level III: • Aliyun Linux 2 Baseline for China classified protection of cybersecurity-Level III • CentOS Linux 6 Baseline for China classified protection of cybersecurity-Level III • CentOS Linux 7 Baseline for China classified protection of cybersecurity-Level III • Debian Linux 8 Baseline for China classified protection of cybersecurity-Level III • Redhat Linux 6 Baseline for China classified protection of cybersecurity-Level III • China's Level 3 Protection of Cybersecurity - Red Hat Enterprise Linux 7 Compliance Baseline Check • SUSE Linux 10 Baseline for China classified protection of cybersecurity-Level III • SUSE Linux 11 Baseline for China classified protection of cybersecurity-Level III • SUSE Linux 12 Baseline for China classified protection of cybersecurity-Level III • Ubuntu 14 Baseline for China classified protection of cybersecurity-Level III • Linux Ubuntu 16 Baseline for China classified protection of cybersecurity-Level III • Windows 2008 R2 Baseline for China classified protection of cybersecurity-Level III • Windows 2012 R2 Baseline for China classified protection of cybersecurity-Level III • Windows 2016/2019 R2 Baseline for China classified protection of cybersecurity-Level III
Weak password	Weak password - Linux system login weak password baseline
	Weak password - SQL Server DB login weak password baseline
	Weak password - PostgreSQL DB login weak password baseline
	Weak password - Windows system login weak password baseline
	Weak password - Ftp login weak password baseline
	Weak password - Mysql DB login weak password baseline
Middleware	Alibaba Cloud Standard - IIS 8 Security Baseline Check
	Alibaba Cloud Standard-Apache Tomcat Security Baseline
	Alibaba Cloud Standard - Apache Security Baseline Check
	Alibaba Cloud Standard - Nginx Security Baseline Check

27.1.6.3.2.2. Configure baseline check policies

This topic describes how to add, modify, and delete baseline check policies and how to set baseline check levels.

Context

By default, the baseline check feature uses the **default policy** to check the baseline security of assets. You can also customize baseline check policies based on your business requirements, for example, to check the compliance with China classified protection of cybersecurity-Level II.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Baseline Check**.

3. In the upper-right corner of the page that appears, click **Manage Policies**. In the **Manage Policies** pane, add, modify, or delete a baseline check policy, or modify the default policy.
   - In the upper-right corner of the pane, click **Create Policy** to customize a baseline check policy. Then, click **Ok**.

Parameter	Description
Policy Name	Enter a policy name.
Schedule	Select a time interval for scheduled scan tasks from: 1 Day(s), 3 Day(s), 7 Day(s), and 30 Day(s), which represent every second day, every fourth day, every eighth day, and every thirty-first day. You can also select a time period for scheduled scan tasks from: 00:00 to 06:00, 06:00 to 12:00, 12:00 to 18:00, and 18:00 to 24:00.
Check Items	Select the baseline items that need to be checked under these categories: High risk exploit, CIS and China's Protection of Cybersecurity, Best security practices, and Weak password.
Servers	Select the asset groups to which you want to apply this policy. **Note** Newly purchased servers belong to **Default** under **Asset Groups**. To apply this policy to new servers, select **Default**.

   - Click **Edit** or **Delete** next to the target policy to modify or delete it.

     Note You cannot restore a policy after it is deleted.

   - Click **Edit** in the **Actions** column next to the **Default** policy to modify the asset groups to which the default policy is applied.

     Note You cannot delete the default policy or modify the check items of the default policy. You can only modify the asset groups to which the default policy is applied.

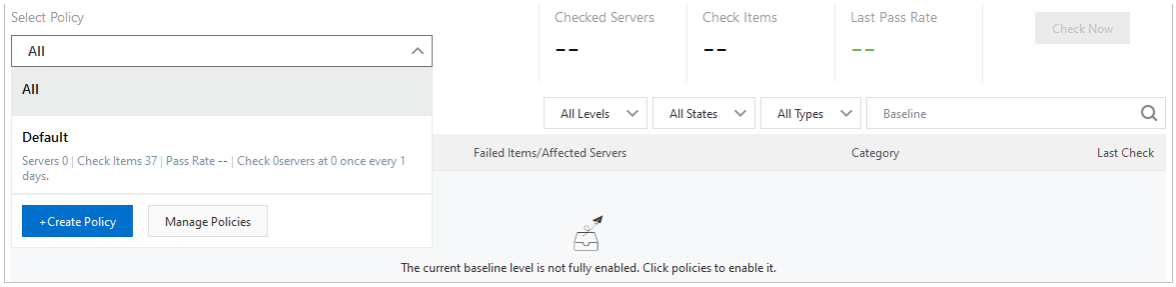
   - In the lower part of the **Manage Policies** pane, set the baseline check level to High, Medium, and Low.
4. Click **Ok**.

27.1.6.3.2.3. View baseline check results and manage failed check items

The Apsara Stack Security console provides detailed baseline check results and suggestions on how to manage failed check items. This topic describes how to view baseline check results and manage failed check items in the Apsara Stack Security console. The check results include affected assets, checked items, and the suggestions.

View the summary of baseline check results

1. Log on to Apsara Stack Security Center.
2.
3. In the upper part of the **Baseline Check** page, view the summary of baseline check results. You can filter data by policy.



You can select a policy from the **Select Policy** drop-down list to view the following information:

- **Checked Servers**: The number of servers on which the baseline check runs. These servers are specified in the selected baseline check policy.
- **Check Items**: The number of **Check Items** specified in the selected baseline check policy.
- **Last Pass Rate**: The pass rate of the last baseline check.

  If the number under **Last Pass Rate** is green, the pass rate of the checked servers is high. If this number is red, a large number of failed check items have been detected on the checked servers. We recommend that you view the check result details and manage the failed check items.

View all check items

1. Log on to Apsara Stack Security Center.
2.
3. Select **All** from the **Select Policy** drop-down list.
   The **Baseline Check** page displays check result details, including **Baseline**, **Checked Item**, **Failed Items/Affected Servers**, **Category**, and **Last Check**.

> **Note** You can also select a baseline check policy from the **Select Policy** drop-down list to view the check items specified in this policy.

View details of a check item

1. Log on to Apsara Stack Security Center.
2.
3. In the **Baseline** column, click the target check item to view its details.
4. On the details page, manage the failed check items.
   - Find the target asset and click **View** in the **Actions** column to open the **At-Risk Items** pane.
   -

View failed check items

1. Open the check item details page. Find the target asset and click **View** in the **Actions** column to view failed check items.
   You can view the check items of the asset and the statuses of the check items (**Passed** or **Failed**).
2.

> **Note** We recommend that you follow the suggestions to manage **Failed** check items at the earliest opportunity, especially the high-risk check items.

Manage failed check items

In the At-Risk Items pane, manage failed check items as required.

- **Add a check item to the whitelist**

If you want to disable alerts for a check item, click **Whitelist** to add the check item to the whitelist. Check items in the whitelist do not trigger alerts.

> **Note** You can also select multiple check items and click **Whitelist** in the lower-left corner to add the check items to the whitelist at a time.

- **Remove a check item from the whitelist**

  If you want to enable alerts for a check item in the whitelist, you can click **Remove** to remove the check item from the whitelist. You can remove one or more check items from the whitelist at a time. After a check item is removed from the whitelist, the check item triggers alerts again.

- **Verify a fixed check item**

  If you do not manually perform the verification, Apsara Stack Security automatically verifies the check item based on the detection interval specified in the policies.

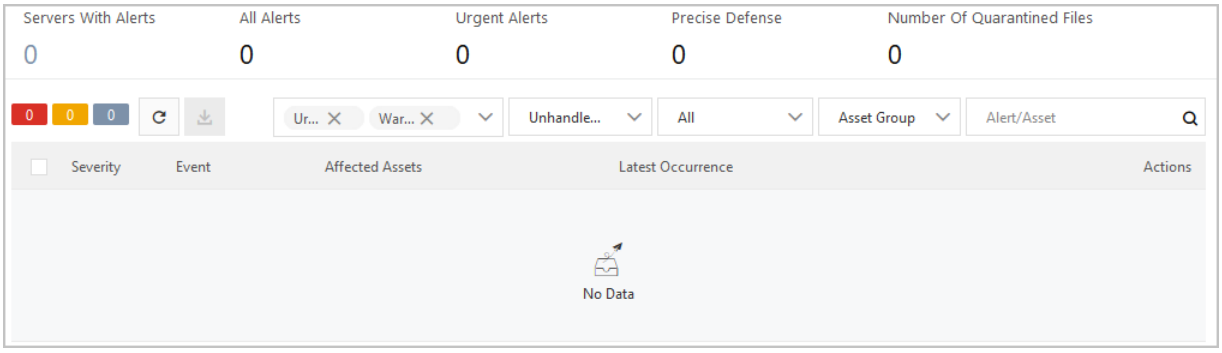
27.1.6.4. Intrusion prevention

27.1.6.4.1. Intrusion events

27.1.6.4.1.1. Intrusion event types

If the Server Security feature detects sensitive file tampering, webshells, unusual logons, suspicious processes, and malicious processes, it generates alerts. Based on these alerts, you can monitor the security status of your assets and handle potential threats at the earliest opportunity.

Apsara Stack Security Center provides statistics based on enabled alerts and defense items. This allows you to understand enabled and disabled defense items. You can view statistics on alerts and information about defense items. To achieve this purpose, choose **Server Security > Intrusion Prevention > Intrusions**.



Alert types

The following table describes the defense items.

Alert	Description
Suspicious Process	Detects whether suspicious processes exist.

Alert	Description
Webshell	Uses engines developed by Alibaba Cloud to scan common webshell files. Apsara Stack Security Center supports scheduled scan tasks, provides real-time protection, and quarantines webshell files. • Apsara Stack Security Center scans the entire web directory early in the morning on a daily basis. A change made to files under the web directory triggers dynamic detection. • You can specify the assets on which Apsara Stack Security Center scans for webshells. • You can quarantine, restore, and ignore detected Trojan files.
Unusual Logon	Detects unusual logons to your servers. You can specify approved logon IP addresses, time periods, and accounts. Logons from unapproved IP addresses, accounts, or time periods trigger alerts. You can manually add approved logon locations or configure the system to automatically update approved logon locations. You can also specify assets on which alerts are triggered when unusual logon locations are detected. Apsara Stack Security Center can detect the following logon events: • Logons to Elastic Compute Service (ECS) instances from unapproved IP addresses • Logons to ECS instances from unapproved locations • Execution of unusual commands after logons to ECS instances by using Secure Shell (SSH) • ECS instances passwords cracked due to brute-force attacks based on the SSH protocol
Sensitive File Tampering	Checks whether sensitive files on your servers are maliciously modified, such as tampering of pre-loaded configuration files in Linux shared libraries.
Malicious Process	Dynamically scans your servers based on the anti-virus mechanism of Alibaba Cloud and Apsara Stack Security Center, and generates alerts if viruses are detected. Process information is collected by Apsara Stack Security Center and uploaded to Alibaba Cloud. You can manage detected viruses in the Apsara Stack Security Center console. Apsara Stack Security Center can detect the following malicious activities and processes: • Visiting malicious IP addresses • Mining programs • Self-mutating Trojans • Malicious programs • Trojans
Unusual Network Connection	Detects disconnections or unusual network connections.
Suspicious Account	Detects logons to your assets from unapproved accounts.
Application intrusion event	Detects intrusion activities that use system application components.
Precision defense	The **Virus Removal** feature provides precise defenses against a majority of ransomware, distributed denial-of-service (DDoS) Trojans, mining programs, Trojan programs, malicious processes, webshells, and worms.
Cloud threat detection	Detects threats in other Alibaba Cloud services.

Alert	Description
Persistence	Detects suspicious scheduled tasks on servers and generates alerts when persistent threats against the servers are detected.
Web Application Threat Detection	Detects server intrusions that use web applications.
Malicious scripts	Checks whether the system services of your assets are attacked or modified by malicious scripts. Alerts are generated if potential script attacks are detected.
Other	Detects other types of attacks, such as DDoS attacks.

27.1.6.4.1.2. View and handle detected alert events

This topic describes how to view and handle detected alert events on the Intrusions page.

Background information

After alert events are detected, they are displayed on the Intrusions page. You can choose **Server Security > Intrusion Prevention > Intrusions** to go to the Intrusions page.

If the alert events are not handled, they are displayed in the **Unhandled Alerts** list on the **Intrusions** page. After the alert events are handled, the status changes from **Unhandled Alerts** to **Handled**.

> **Note** Apsara Stack Security Center retains the records of **Unhandled Alerts** and **Handled** on the **Intrusions** page. By default, the records of **Unhandled Alerts** are displayed.

View detected alert events

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Intrusion Prevention > Intrusions**.
3. In the intrusions list, search for or view detected intrusion events, alert events, and relevant details.

Handle alert events

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Intrusion Prevention > Intrusions**.
3. On the **Intrusions** page, find the specific alert event and click **Processing** in the **Actions** column. In the dialog box that appears, configure parameters and click **Process Now**.

   > **Note** If the alert event contains multiple correlated exceptions, click **Processing**. On the details page that appears, you can handle different alert events.

   - **Ignore**: If you ignore the alert event, the status of the alert event changes to **Handled**. The system no longer generates alerts for the event.
   - **Whitelist**: If the alert event is a false positive, you can add the alert event to the whitelist. After you add the alert event to the whitelist, the status of the alert event changes to **Handled**. Server Guard no longer generates alerts for the event. In the **Handled** alert list, you can click **Cancel whitelist** to remove the specific alert event from the whitelist.

     > **Note** A false positive represents that the system has generated a false alert on a normal process. Common false positives include **suspicious processes that send TCP packets**. The false positive notifies you that suspicious scans other devices have been detected on your servers.

- **Batch unhandled**: This method handles multiple alerts at a time. Before you handle multiple alert events at a time, we recommend that you view the details of the alert events.

4. (Optional)If you confirm that one or more alert events are false positives or need to be ignored, go to the **Intrusions** page, select the specific alert events, and then click **Ignore Once** or **Whitelist**.

Export alert events

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Intrusion Prevention > Intrusions**.
3. In the upper-left corner of the **Intrusions** page, click the icon to export the alert list.

   After the alert list is exported, the **Done** message appears in the upper-right corner.
4. In the **Done** notification of the **Alerts** page, click **Download**.
   The alert list is downloaded to your computer.

27.1.6.4.1.3. View exceptions related to an alert

Server Guide supports automatic analysis of exceptions related to an alert. You can click an alert name on the alert list to view and manage all exceptions related to this alert and view the results of automatic attack tracing.

Context

- Security Center automatically associates alerts with exceptions in real time to detect potential threats.
- Exceptions related to an alert are listed in chronological order. This allows you to analyze and handle the exceptions to improve the emergency response mechanism of your system.
- An automatically correlated alert is identified by the icon.

Procedure

1. Log on to Apsara Stack Security Center.
2.
3. On the Intrusions page, click the target **alert name**. The alert details page appears.
4. On the alert details page, view the details and related exceptions of the alert and manage the exceptions.
   - View alert details

     You can view the assets affected by this alert, the first and latest occurrence time of the alert, and the details of the related exceptions.
   - View affected assets

     Click the name of an **affected asset** to view the details of the asset. These details include alerts, vulnerabilities, baseline risks, and asset fingerprints.
   - View and manage **related exceptions**

     In the **Related Exceptions** section, you can view the details and recommended solutions of all exceptions related to this alert.
     - Click **Note** to the right of an exception name to add notes for the exception.
     - Click the icon to the right of a note to delete the note.

27.1.6.4.1.4. Use the file quarantine function

Sever Guide can quarantine malicious files. Quarantined files are listed in the quarantine box on the Intrusions page. The system automatically deletes a quarantined file 30 days after it is quarantined. You can restore a quarantined file with a few clicks before it is deleted. This topic describes how to view quarantined files and remove files from the quarantine box.

Procedure

1. Log on to Apsara Stack Security Center.
2.
3. In the upper-right corner of the **Intrusions** page, click **Quarantine**.

   You can perform the following operations in the **Quarantine** pane:

   - View information about quarantined files. The information includes server IP addresses, directories that store the files, file status, and time of the last modification.
   - Click **Restore** in the **Actions** column to remove a file from the quarantine box. The restored file appears in the alert list again.

27.1.6.4.1.5. Configure security alerts

This topic describes how to configure security alerts, which allow you to specify approved logon locations and customize web directories to scan.

Context

Server Guard supports advanced logon settings and security alerts. You can configure more fine-grained logon detection rules. For example, you can specify approved logon IP addresses, logon time, and logon accounts to block unauthorized requests sent to your assets.

Procedure

1. Log on to Apsara Stack Security Center.
2.
3. In the upper-right corner, click **Settings**.

   Configure parameters on different tabs.

   - **Add an approved logon location**
     a. Click **Management** to the right of **Login Location**.
     b. Select the logon location that you want to add, and select the servers that allow logons from the added location.
     c. Click **OK**.

     Server Guard allows you to **edit** and **delete** approved logon locations.

     - Find the specific logon location and click **Edit** on the right side to change the servers that allow logons from this location.
     - Find the specific logon location and click **Delete** on the right side to delete the logon location.

   - **Configure advanced logon settings**

     > **Note** When you configure advanced logon settings, you can specify the IP addresses, accounts, and time ranges that are allowed for logons to your assets. After the advanced logon settings are configured, Server Guard sends you alerts if your assets receive unauthorized logon requests. The procedure of configuring advanced logon settings is similar to that of configuring **Login Location**. You can **add**, **edit**, and **delete** advanced logon settings in the similar way.

- Turn on or turn off Uncommon IP Alert to the right of **Common Login IPs**. If your assets receive logon requests from unauthorized IP addresses, alerts are triggered.
- Turn on or turn off Uncommon Time Alert to the right of **Common Login Time**. If your assets receive logon requests at unauthorized time periods, alerts are triggered.
- Turn on or turn off Uncommon Account Alert to the right of **Common Login Accounts**. If your assets receive logon requests from unauthorized accounts, alerts are triggered.

- **Add web directories to scan**

  Server Guard automatically scans web directories of data assets in your servers and runs dynamic and static scan tasks. You can also manually add other web directories of your servers.

  a. On the right of **Add Scan Targets**, click **Management**.
  b. Specify a valid web directory and select the servers on which the specified web directory is scanned.

  > **Note** To ensure the performance and efficiency, do not specify a root directory.

  c. Click **OK**.

27.1.6.4.1.6. Virus removal

The virus removal feature provided by Server Guard is integrated with major antivirus engines worldwide. It detects viruses against large amounts of threat intelligence data provided by Alibaba Cloud. Virus removal also provides an exception detection module designed by Alibaba Cloud to detect viruses based on machine learning and deep learning. This way, virus removal can provide full-scale and dynamic antivirus protection to safeguard your servers.

The cloud threat detection feature scans hundreds of millions of files on a daily basis and protects millions of servers on the cloud.

Detection capabilities

The virus removal feature uses the Server Guard client to collect process information, and then scans the retrieved data for viruses in the cloud. If a malicious process is detected, you can stop the process and quarantine the related files.

- **Deep learning engine developed by Alibaba Cloud**: The deep learning engine is built on deep learning technology and a large number of attack samples. The engine detects malicious files on the cloud and automatically identifies potential threats to supplement traditional antivirus engines.
- **Cloud sandbox developed by Alibaba Cloud**: The cloud sandbox feature allows you to simulate cloud environments and monitor attacks launched by malicious samples. The cloud sandbox feature automatically detects threats and offers dynamic analysis and detection capabilities based on big data analytics and machine learning modeling techniques.
- **Integration with major antivirus engines**: The cloud threat detection feature is integrated with major antivirus engines and updates its virus library in real time.
- **Threat intelligence detection**: The cloud threat detection feature works with the exception detection module to detect malicious processes and operations based on threat intelligence data provided by Alibaba Cloud Security.

Detectable virus types

The cloud threat detection feature is developed based on the security technologies and expertise of Alibaba Cloud. The feature provides end-to-end security services, including threat intelligence collection, data masking, threat identification, threat analysis, and malicious file quarantine and restoration. You can quarantine and restore files that contain viruses in the Security Center console.

The cloud threat detection feature can detect the following types of viruses.

Virus	Description
Mining program	A mining program consumes server resources and mines cryptocurrency without authorization.
Computer worm	A computer worm uses computer networks to replicate itself and spread to a large number of computers within a short period of time.
Ransomware	Ransomware, such as WannaCry, uses encryption algorithms to encrypt files and prevent users from accessing the files.
Trojan	A trojan is a program that allows an attacker to access information about servers and users, gain control of the servers, and consume system resources.
DDoS trojan	A DDoS trojan hijacks servers and uses zombie servers to launch DDoS attacks, which interrupts your service.
Backdoor	A backdoor is a malicious program injected by an attacker. Then, the attacker can use the backdoor to control the server or launch attacks.
Computer virus	A computer virus inserts malicious code into normal programs and replicates the code to infect the whole system.
Malicious program	A malicious program may pose threats to system and data security.

Benefits

- **Self-developed and controllable**: The cloud threat detection feature is based on deep learning, machine learning, and big data analytics with a large number of attack and defense practices. The feature uses multiple detection engines to dynamically protect your assets against viruses.
- **Lightweight**: The cloud threat detection feature consumes only 1% of CPU resources and 50 MB of memory.
- **Dynamic**: The cloud threat detection feature dynamically retrieves startup logs of processes to monitor the startup of viruses.
- **Easy to manage**: You can manage all servers and view their status at any time in the Security Center console.

Threat detection limits

Apsara Stack Security Center allows you to process security alerts, scan for vulnerabilities, analyze attacks, and check security settings in the Security Center console. Apsara Stack Security Center can analyze alerts and automatically trace attacks. This allows you to protect your assets. Apsara Stack Security Center supports a wide range of protection features. We recommend that you also install the latest system patches on your server. We also recommend you to use security services, such as Cloud Firewall and Web Application Firewall (WAF), to better protect your assets against attacks.

> **Note** Attacks and viruses are evolving, and your business environments vary. Security breaches may occur. We recommend that you use the alerting, vulnerability detection, baseline check, and configuration assessment features provided by Apsara Stack Security Center to better protect your assets against attacks.

27.1.6.4.2. Website tamper-proofing

27.1.6.4.2.1. Overview

Tamper protection monitors website directories in real time, restores modified files or directories, and protects websites from trojans, hidden links, and uploads of violent and illicit content.

Background information

To make illegal profits or conduct business attacks, attackers exploit vulnerabilities in websites to insert illegal hidden links and tamper with the websites. Defaced web pages affect normal user access and may lead to serious economic losses, damaged brand reputation, or political risks.

Tamper protection allows you to add Linux and Windows processes to the whitelist and update protected files in real time.

How tamper protection works

The Security Center agent automatically collects the list of processes that attempt to modify files in the protected directories of the protected servers. It identifies unusual processes and file changes in real time and blocks unusual processes.

The alert list is displayed on the Tamper Protection page. You can view unusual file changes, the corresponding processes, and the number of attempts made by each process in the alert list. If a file is modified by a trusted process, you can add the process to the whitelist. After the process is added to the whitelist, tamper protection no longer blocks the process. In scenarios where the content of websites, such as news and education websites, is frequently modified, the whitelist saves you the effort of frequently enabling and disabling tamper protection.

Versions of operating systems and kernels supported by tamper protection

OS	Supported operating system version	Supported kernel version
Windows	Windows Server 2008 and later	All versions
CentOS	6.5, 6.6, 6.7, 6.8, 6.9, 6.10, 7.0, 7.1, 7.2, 7.3, 7.4, 7.5, and 7.6	• 2.6.32-x • 3.10.0-x
Ubuntu	14, 16, and 18	• 3.13.0-32-generic • 3.13.0-86-generic • 4.4.0-62-generic • 4.4.0-63-generic • 4.4.0-93-generic • 4.4.0-151-generic • 4.4.0-117-generic • 4.15.0-23-generic • 4.15.0-42-generic • 4.15.0-45-generic • 4.15.0-52-generic

Note

- The preceding table lists kernel versions supported by tamper protection. Servers that use an unsupported kernel version cannot use tamper protection. Make sure that your server uses a supported kernel version. If a kernel version is not supported, you must upgrade it to a supported version. Otherwise, you cannot add processes to the whitelist.
- Before you upgrade the server kernel, back up your asset data.

27.1.6.4.2.2. Configure tamper protection

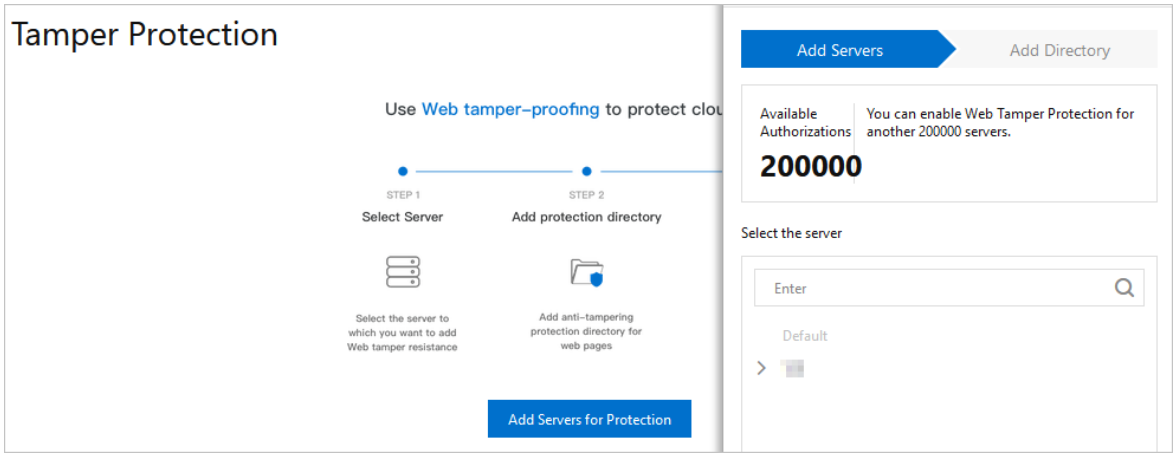
The Server Security feature allows you to configure tamper protection for web pages.

Limits

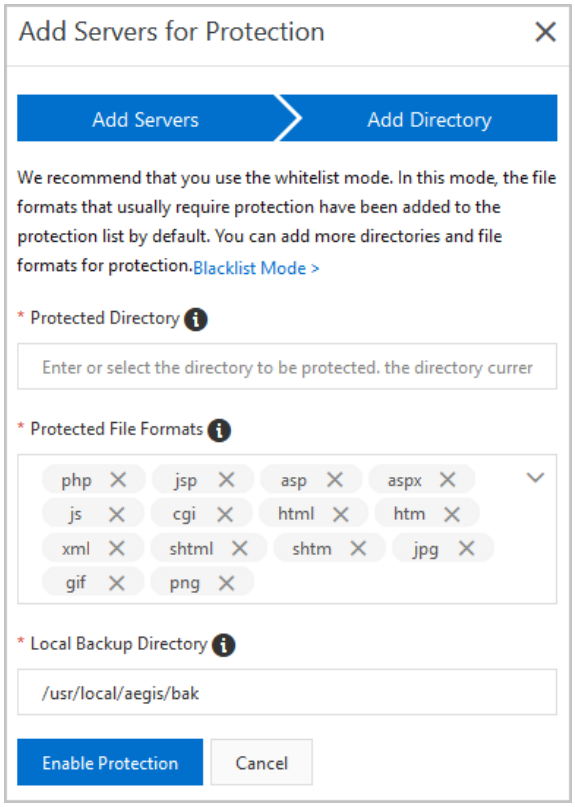
- For each server, you can add a maximum of 10 directories for protection.
- The protected directories of a Windows server must meet the following requirements: The maximum size of each directory is no more than 20 GB. Each directory contains a maximum of 2,000 folders. The maximum directory level is 20. The maximum size of each file is 3 MB.
- The protected directories of a Linux server must meet the following requirements: The maximum size of each directory is no more than 20 GB. Each directory contains a maximum of 3,000 folders. The maximum directory level is 20. The maximum size of each file is 3 MB.
- Before you add a directory for protection, make sure that the directory level, the number of folders, and the directory size meet the preceding requirements.
- We recommend that you exclude file formats that do not require protection, such as *.log*, *.png*, *.jpg*, *.mp4*, *.avi*, and *.mp3*. Separate multiple file formats with semicolons (;).

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, click **Server Security** and choose **Intrusion Prevention > File Tamper Protection**.
3.
4. On the **Tamper Protection** page, click **Add Servers for Protection**.
5. In the **Add Servers for Protection** dialog box, select a server that you want to protect.



6. Click **Next** to go to the **Add Directory** step.
7. In the **Add Directory** step, configure the following parameters:



Select the protection mode. You can select the **Whitelist Mode** or **Blacklist Mode**. In whitelist mode, tamper protection is enabled for the specified directories and file formats. In blacklist mode, tamper protection is enabled for the subdirectories, file formats, and files that are not specified. The whitelist mode is selected by default.

- In whitelist mode, configure the following parameters.

Parameter	Description
Protected Directory	Enter the path of the directory that you want to protect. **Note** Servers that run Linux and Windows operating systems use different path formats. Enter a valid directory path based on your operating system type.
Protected File Formats	Select file formats from the drop-down list, such as *.js*, *.html*, *.xml*, and *.jpg*.
Local Backup Directory	The default path where backup files of the protected directories are stored. By default, Security Center assigns */usr/local/aegis/bak* to servers that run the Linux operating system and *C:\Program Files (x86)\Alibaba\Aegis\bak* to servers that run the Windows operating system. You can change the default path as needed.

- In blacklist mode, configure the following parameters.

Parameter	Description
Protected Directory	Enter the path of the directory that you want to protect.

Parameter	Description
Excluded Sub-Directories	Enter the subdirectories that do not require tamper protection. Click **Add Sub-Directory** to enter more subdirectories. Security Center does not provide tamper protection for files under the excluded sub-directories.
Excluded File Formats	Select the formats of files that do not require tamper protection. Supported formats include **log**, **txt**, and **ldb**. Security Center does not provide tamper protection for the files in the excluded formats.
Excluded Files	Enter the path of the file that does not require tamper protection. You can click **Add File** to add more files. Security Center does not provide tamper protection for the excluded files.
Local Backup Directory	The default path where backup files of the protected directories are stored. By default, Security Center assigns */usr/local/aegis/bak* to servers that run the Linux operating system and *C:\Program Files (x86)\Alibaba\Aegis\bak* to servers that run the Windows operating system. You can change the default path as needed.

8. Click **Enable Protection**.
   After you enable tamper protection for a server, it is displayed in the server list on the **Tamper Protection** page.

   > **Note** By default, tamper protection is in the **Not Initiated** state for newly added servers. To enable tamper protection, you must turn on the **On** switch on the **Tamper Protection** page for the target server.

9. In the server list on the Tamper Protection page, find the target server and turn on the switch in the **Protection** column to enable tamper protection for the server.

   > **Note** By default, tamper protection is in the **Not Initiated** state for newly added servers. You must enable tamper protection on the **Tamper Protection** page for the target server.

   After tamper protection is enabled, the protection state changes to **Running**.

   > **Note** If the protection state of a server is **Exception**, click **Exception** in the Status column. A message that indicates the causes appears. Click **Retry** in the message.

What to do next

After you enable tamper protection for a server, you can go to the **Overview** page, and select tamper protection in the event type drop-down list to view the alerts on tampering events.

> **Note**
>
> Tamper protection does not take effect immediately after you configure the protected directory, and you can still write files to the directory. In this case, you must go to the **Management** page, disable **Protection** for the server where the directory is located, and then enable **Protection** again.

Handle abnormal protection states

Protection state	Description	Suggestion
Initializing	Web tamper protection is being initialized.	If this is your first time enabling tamper protection for a server, the protection status becomes **Initializing**. It takes a few seconds to enable tamper protection.
Running	Tamper protection is enabled and running as expected.	-
Exception	An error occurred when tamper protection was enabled.	Click **Exception** in the Status column to view the exception cause and click Retry.
Not Initiated	Tamper protection is disabled.	You must turn on the **On** switch to enable tamper protection.

27.1.6.4.2.3. View the protection status

This topic describes how to view the status of tamper protection for your assets.

Context

The tamper protection feature monitors changes of directories and files in real time and blocks suspicious file changes. To view the status and details of tamper protection in the Apsara Stack Security console, click **Server Security** and choose **Intrusion Prevention > File Tamper Protection**. The details include:

- Overview

  On the Overview page, you can view the total number of changed files, protected servers, and protected directories on the current day and during the last 15 days.

- Distribution of protected file types

  Protected file types include .txt, .png, .msi, and .zip. You can also add more types of files for protection as needed.

  > **Note** All types of files can be added for tamper protection.

- Top five files with the largest number of changes

  This module shows the names and paths of the five files with the largest number of changes in the last 15 days.

- Details of tamper protection alerts

  The tamper protection feature helps you block all suspicious changes to directories and files on your assets. On the alert details page, you can view the alerts of these changes, including the severity, alert name, affected assets, paths of files with suspicious changes, and protection status.

> Note
> - If the number of alerts exceeds 100, we recommend that you process these alerts at your earliest opportunity.
> - Only the alerts at the **Warning** level are displayed in the console.
> - Only alerts in the **Isolation successful** state are displayed. This indicates that the tamper protection feature has blocked the suspicious processes that attempted to make unauthorized file changes.

27.1.6.4.3. Configure the Virus Removal feature

The Virus Removal feature of Server Guard allows you to customize virus and webshell detection settings.

Virus detection

The Virus Removal feature can automatically quarantine common Internet viruses, such as mainstream trojans, ransomware, mining programs, and DDoS trojans. Apsara Stack Security experts test and verify all automatically quarantined viruses to ensure a minimum false positive rate.

If automatic quarantine is disabled, Server Guard generates alerts when viruses are detected. You must manually manage detected viruses in the console. We recommend that you enable automatic quarantine to better safeguard your servers.

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, click **Server Security** and choose **Intrusion Prevention > Virus Removal**.
3. In the **Anti-virus** section, click **Manage**.
4. In the **Configure Servers for Virus Detection** dialog box, select the servers for which you want to enable the Virus Removal feature.
5. Click **OK**.
6. In the **Anti-virus** section, turn on **Virus Blocking** to enable virus blocking.

   After virus blocking is enabled, Server Guard automatically quarantines detected viruses. Quarantined viruses are listed on the Overview page. You can select the **Precision Defense** type to filter quarantined viruses.

Webshell detection

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, click **Server Security** and choose **Intrusion Prevention > Virus Removal**.
3. Configure servers for webshell detection.
   i. In the **Webshell Detection** section, click **Manage**.
   ii. Select the servers for which you want to enable webshell detection.
   iii. Click **OK** to complete the configuration.

27.1.6.5. Log retrieval

27.1.6.5.1. Log retrieval overview

The log retrieval function provided by Server Security allows you to manage logs scattered in various systems of Apsara Stack in a centralized manner, so that you can easily identify the causes of issues that occur on your servers.

The log retrieval function supports storage of logs for 180 days and query of logs generated within 30 days.

Benefits

The log retrieval function provides the following benefits:

- **End-to-end log retrieval platform**: Allows you to retrieve logs of various Apsara Stack services in a centralized manner and trace issues easily.
- **Cloud-based SaaS service**: Allows you to query logs on all servers in Apsara Stack without additional installment and deployment.
- Supports TB-level data retrieval. It also allows you to add a maximum of 50 inference rules (Boolean expressions) in a search condition and obtain full-text search results within several seconds.
- Supports a wide range of log sources.
- Supports log shipping, which allows you to import security logs to Log Service for further analysis.

Scenarios

You can use log retrieval to meet the following requirements:

- **Security event analysis**: When a security event is detected on a server, you can retrieve the logs to identify the cause and assess the damage and affected assets.
- **Operation audit**: You can audit the operation logs on a server to identify high-risk operations and serious issues in a meticulous way.

Supported log types

Log types

Log type	Description
Logon history	Log entries about successful system logons
Brute-force attack	Log entries about system logon failures that are generated during brute-force attacks
Process snapshot	Log entries about processes on a server at a specific time
Listening port snapshot	Log entries about listening ports on a server at a specific time
Account snapshot	Log entries about account logon information on a server at a specific time
Process initiation	Log entries about process initiation on a server
Network connection	Log entries about active connections from a server to external networks

27.1.6.5.2. Query logs

This topic describes how to search for and view server logs.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Log Retrieval**.
3. Specify search conditions.

Search condition	Description
Log source	The supported log source. For more information, see Log sources.

Search condition	Description
Field	The field that is supported by the specified log source. For more information, see Log sources.
Keyword	The keyword of the field that you want to search for.
Logical operator	The equality operator.
+	The inference rules in a search condition for a log source.
Add conditions	The search conditions for different log sources.

4. Click **Search** and view the search result.
   - **Reset**: Click **Reset** to clear the search condition configurations.
   - **Save Search**: Click **Save Search** to save the search condition configurations for future use.
   - **Saved Searches**: Click **Saved Searches** to select and apply a search condition configuration that has been saved.

27.1.6.5.3. Supported log sources and fields

This topic describes the log sources and fields that are supported by the log retrieval feature.

The log retrieval feature allows you to query the following types of log sources. You can click a log source link to view the fields that can be retrieved.

Log source	Description
Logon history	Log entries about successful system logons.
Logs of brute-force attacks	Log entries about failed system logons during brute-force attacks.
Process snapshot logs	Log entries about processes on a server at a specific point in time .
Logs of listening port snapshots	Log entries about listening ports on a server at a specific point in time.
Account snapshot logs	Log entries about account-based logons on a server at a specific point in time.
Process startup logs	Log entries about process startups on a server.
Network connection logs	Log entries about active connections from a server to the Internet.

Logon history

You can query the logon history by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.
IP	string	The IP address of the server.
warn_ip	string	The source IP address for the logon.
warn_port	string	The logon port.

Field	Data type	Description
warn_user	string	The username for the logon.
warn_type	string	The logon type.
warn_count	string	The number of logon attempts.

Logs of brute-force attacks

You can query logs about brute-force attacks by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.
IP	string	The IP address of the server.
warn_ip	string	The source IP address of the attack.
warn_port	string	The target port.
warn_user	string	The target username.
warn_type	string	The type.
warn_count	string	The number of brute-force attack attempts.

Process startup logs

You can query process startup logs by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.
IP	string	The IP address of the server.
pid	string	The ID of the process.
groupname	string	The user group.
ppid	string	The ID of the parent process.
uid	string	The ID of the user.
username	string	The username.
filename	string	The file name.
pfilename	string	The name of the parent process file.
cmdline	string	The command line.
filepath	string	The path of the process file.
pfilepath	string	The path of the parent process file.

Logs of listening port snapshots

You can query logs about listening port snapshots by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.
IP	string	The IP address of the server.
src_port	string	The listening port.
src_ip	string	The listening IP address.
proc_path	string	The path of the process file.
pid	string	The ID of the process.
proc_name	string	The name of the process.
proto	string	The protocol.

Account snapshot logs

You can query account snapshot logs by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.
IP	string	The IP address of the server.
perm	string	Indicates whether the user has root permissions.
home_dir	string	The home directory.
warn_time	string	The time at which a password expiration notification is sent.
groups	string	The group to which the user belongs.
login_ip	string	The IP address of the last logon.
last_chg	string	The time at which the password was last changed.
shell	string	The Linux shell command.
domain	string	The Windows domain.
tty	string	The logon terminal.
account_expire	string	The time at which the account expires.
passwd_expire	string	The time at which the password expires.
last_logon	string	The last logon time.

Field	Data type	Description
user	string	The username.
status	string	The account status. Valid values: • 0: disabled • 1: normal

Process snapshot logs

You can query process snapshot logs by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.
IP	string	The IP address of the server.
path	string	The path of the process file.
start_time	string	The time at which the process was started.
uid	string	The ID of the user.
cmdline	string	The command line.
pname	string	The name of the parent process.
name	string	The name of the process.
pid	string	The ID of the process.
user	string	The username.
md5	string	The MD5 hash of the process file. If the size of the process file exceeds 1 MB, the system does not calculate the MD5 hash value of the process file.

Network connection logs

You can query network connection logs by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.
IP	string	The IP address of the server.
src_ip	string	The source IP address.
src_port	string	The source port.
proc_path	string	The path of the process file.
dst_port	string	The destination port.

Field	Data type	Description
proc_name	string	The name of the process.
dst_ip	string	The destination IP address.
status	string	The status.

27.1.6.5.4. Logical operators

The log retrieval feature supports multiple search conditions. You can add multiple logical operators to one search condition for one log source, or combine multiple search conditions for several log sources by using different logical operators. This topic describes the logical operators that are supported in log retrieval. Examples are provided to help you understand these operators.

The following table describes the logical operators that are supported in log retrieval.

Logical operators

Logical operator	Description
and	Binary operator. This operator is in the format of `query 1 and query 2`, which indicates the intersection of the query results of `query 1` and `query 2`. **Note** If no logical operators are used for multiple keywords, the default operator is AND.
or	Binary operator. This operator is in the format of `query 1 or query 2`, which indicates the union of the query results of `query 1` and `query 2`.
not	Binary operator. This operator is in the format of `query 1 not query 2`, which indicates the results that match `query 1` but do not match `query 2`. This format is equivalent to `query 1 - query 2`. **Note** If you use only `not query 1`, the log data that does not contain the query results of `query 1` is returned.

27.1.6.6. Settings

27.1.6.6.1. Install the Server Guard agent

This topic describes how to install the Server Guard agent by specifying parameters.

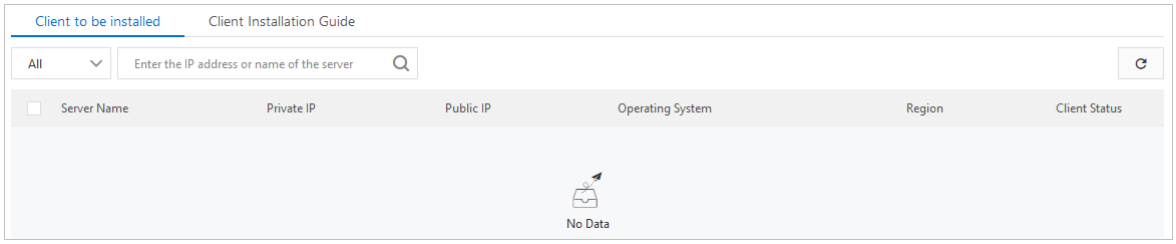
Context

To use the protection services provided by Server Guard, you must install the Server Guard agent on the operating system of your server.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Server Settings > Client Installation**.
3. (Optional)On the Client Installation page, click the **Client to be installed** tab to view the number of servers on which the Server Guard agent is not installed. On this tab, you can also view the relevant information in the server list.

   You can specify the operating system type, IP address, or server name to search for a target server.



4. Click the **Client Installation Guide** tab.
5. Obtain and install the Server Guard agent based on the operating system type of your server.
   - **Windows**
     a. In the left-side pane of the page, click **Click to download** to download the client software package to your computer.
     b. Upload the installation package to your server. For example, you can use an FTP client to upload the package to your server.
     c. Run the installation package on your server as an administrator.

     > **Note** When you install the agent on a server that is not in Alibaba Cloud, you will be prompted to enter the installation verification key. You can find the installation verification key on the Server Guard agent installation page.

   - **Linux**
     a. In the right-side pane of the page, select Alibaba Cloud Server or Non-Alibaba Server.
     b. Select the installation command for your 32-bit or 64-bit operating system and click **Copy** to copy the command.
     c. Log on to your Linux server as an administrator.
     d. Run the installation command on your Linux server to download and install the Server Guard agent.

27.1.6.6.2. Manage protection modes

This topic describes how to manage protection modes for each server to make them more efficient and secure.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Server Security > Server Settings > Protection Mode**.
3. On the Protection Mode page, click **Manage**.

   Configure protection modes for each server.
   - **Business First Mode**: The peak CPU utilization is less than 10%, and the peak memory usage is less than 50 MB.
   - **Protection First Mode**: The peak CPU utilization is less than 20%, and the peak memory usage is less than 80 MB.
4. Click **OK**.

27.1.7. Physical server security

27.1.7.1. Create and grant permissions to a security administrator account

The physical server security feature is used to ensure the security of physical servers on the platform side. This feature requires you to use a dedicated security administrator account for the platform. This topic describes how to create and grant permissions to a security administrator account.

Procedure

1. Log on to the Apsara Uni-manager Management Console as a system administrator.

   For more information, see the **"Log on to the Apsara Uni-manager Management Console"** topic of *Apsara Uni-manager Management Console User Guide*.

2. Create a dedicated organization that is used to manage the security of physical servers, and obtain the primary key of the organization.

   > **Notice** Make sure that the organization is used only to manage the security of physical servers. Do not add Elastic Compute Service (ECS) instances to the organization.

   i. Create the dedicated organization.

      For more information, see **Enterprise Center > Organization Management > Create Organization** in *Apsara Uni-manager Management Console User Guide*.

   ii. Obtain the **primary key** of the newly created organization.

      For more information, see **Enterprise CenterOrganization ManagementObtain the AccessKey pair of an organization** in *Apsara Uni-manager Management Console User Guide*.

3. Create a dedicated account to manage the security of physical servers.

   For more information, see **Enterprise Center > User Management > System User Management > Create User** in *Apsara Uni-manager Management Console User Guide*.

   > **Note** When you create the account, take note of the following points for the **organization** and **role**:
   >
   > - In the **Organization** section, select the organization that is created in the previous step.
   > - In the **Role** section, select **Platform Security Configuration Administrator** and **Security System Configuration Administrator**.

4. Log on to Apsara Stack Security Center by using the newly created account.

   For more information, see Log on to Apsara Stack Security Center.

5. Add the **primary key** of the newly created organization to the protection configuration of physical servers.

   i. In the left-side navigation pane, choose **Security Management Center (SOC) > System Configuration > Global Settings**.

   ii. On the **Global Settings** page, click the **Physical Server Protection** tab.

   iii. Click **Add Account**.

   iv. In the **Add Account** dialog box, specify the **Username** and **Primary Key** parameters.

   - **Username**: Enter the account that you created in Step 3.
   - **Primary Key**: Enter the primary key that you obtained in Step 2.

v. Click **OK**.

Result

After you complete the configuration, the **Physical Server Security** feature appears in the left-side navigation pane of Apsara Stack Security. Then, you can use the account created in this topic to maintain the security of physical servers.

27.1.7.2. View the information on the Overview page

Security administrators can view the security status of all physical servers on the Overview page of Apsara Stack Security.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   > **Note** For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Overview**.
3. On the **Overview** page, view the security status of your physical servers.
   - Security event statistics: displays the numbers of security events on the physical servers. Security events include unusual logons, webshells, and server exceptions.
   - **Events**: displays the trends of security events on the physical servers. A security event is an intrusion event that is detected on a physical server.
   - **Protection Status**: displays the number of physical servers that are protected and the number of offline physical servers.
   - **Recent Important Events**: displays the recent important security events that are detected on the physical servers. You can click a security event to view its details.

27.1.7.3. Physical servers

27.1.7.3.1. Manage physical server groups

This topic describes how to manage physical server groups. To facilitate the security management of physical servers, you can add the physical servers to groups and view their security events by group.

Context

If you do not add a physical server to a server group, the physical server does not belong to a server group by default. If you delete a group, all the physical servers in the group no longer belong to a server group.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   > **Note** For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Physical Servers**.
3. In the left-side group section, manage sever groups.
   - Create a group.

     Click the Add Subgroup icon next to **All Servers** or a specific group, enter a group name, and click **OK**.

> **Note** The system supports a maximum of three levels of groups.

- Modify a group.

  Click the Modify Group Name icon next to the target group, enter a new name, and click **OK**.

- Delete a group.

  Click the Delete icon next to the target group. In the message that appears, click **OK**.

  > **Note** After you delete a group, all servers in the group are automatically moved to the **default** group.

- Sort groups.

  Click **Manage Groups** to sort groups in descending order by priority.

4. Change the server group of specific physical servers.
   i. Select servers from the list on the right.
   ii. Click **Change Group**.
   iii. In the Change Group dialog box that appears, select a group from the drop-down list.
   iv. Click **OK**.

27.1.7.3.2. Manage physical servers

This topic describes how to manage servers. On the Servers page, you can view the status of servers protected by Server Guard.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   > **Note** For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Servers**.
3. (Optional)Search for a server.

   To view the agent status of a server, enter the server IP address in the search bar, and click **Search**. Detailed server information, such as security information, is displayed.

4. View the agent status and detailed security information of the server.

   Click

   in the upper-right corner of the page to select the information columns you want to display. The following table lists the information categories.

Category	Information
Basic information	○ Server IP/Name ○ Tag ○ OS ○ Region
Agent status	Agent Status

Category	Information
Threat prevention	○ Vulnerability ○ Baseline Risk
Intrusion detection	○ Unusual Logons ○ Webshells ○ Suspicious Servers
Server fingerprints	○ Processes ○ Ports ○ Root Accounts/Total Accounts

5. Manage servers.

Action	Description
Change Group	Select servers and click **Change Group** to add the selected servers to a new group.
Modify Tag	Select servers and click **Modify Tag** to modify tags for the servers.
Security Inspection	Select servers and click **Security Inspection** to select the items to be checked.
Delete External Servers	Select **external** servers, and choose **More > Delete External Servers**.
Disable Protection	Select the servers whose agent status is **Online**, and choose **More > Disable Protection**. This temporarily disables protection for these servers to reduce server resource consumption.
Enable Protection	Select the servers whose agent status is **Disable Protection**, and choose **More > Enable Protection**. This enables protection for these servers.

27.1.7.4. Intrusion detection

27.1.7.4.1. Configure policies to identify unusual logons

This topic describes how to configure logon security. You can set approved locations, IP addresses, time periods, and accounts for logons.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   > **Note** For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Intrusion Detection**.
3. Click the **Unusual Logons** tab.
4. Click **Logon Security** in the upper-right corner of the page.
5. Set approved logon locations.

To add an approved logon location, follow these steps:

i. Click **Add**.
ii. Select a logon location from the drop-down list.
iii. Specify the servers on which the selected logon location takes effect.
   - Click **All Servers** to select specific servers.
   - Click **Server Groups** to select servers by group.
iv. Click **OK**.

> **Note** Click **Modify** or **Delete** to modify or delete an approved logon location.

6. Set approved logon IP addresses.

   Turn on the **Disapproved IP Alert** switch. The switch is turned on if it turns green.

   To add an approved logon IP address, follow these steps:

   i. Click **Add**.
   ii. In the **Specify an Approved Logon IP** section, enter an IP address.
   iii. Specify the servers on which the specified IP address takes effect.
      - Click **All Servers** to select specific servers.
      - Click **Server Groups** to select servers by group.
   iv. Click **OK**.

   Click **Modify** or **Delete** to modify or delete an approved logon IP address.

7. Set approved logon time periods.

   Turn on the **Disapproved Time Alert** switch. The switch is turned on if it turns green.

   To add an approved logon time period, follow these steps:

   i. Click **Add**.
   ii. In the **Specify an Approved Logon Duration** section, specify a time period.
   iii. Specify the servers on which the specified logon time period takes effect.
      - Click **All Servers** to select specific servers.
      - Click **Server Groups** to select servers by group.
   iv. Click **OK**.

   Click **Modify** or **Delete** to modify or delete an approved logon time period.

8. Set approved accounts.

   Turn on the **Disapproved Account Alert** switch. The switch is turned on if it turns green.

   To add an approved account, follow these steps:

   i. Click **Add**.
   ii. In the **Specify an Approved Account** section, enter an account.
   iii. Select the servers on which the specified account takes effect.
      - Click **All Servers** to select specific servers.
      - Click **Server Groups** to select servers by group.
   iv. Click **OK**.

   Click **Modify** or **Delete** to modify or delete an approved account.

What's next

After you configure policies to identify unusual logons, you can view and handle unusual logons on the **Unusual Logons** tab. For more information, see Handle unusual logons.

27.1.7.4.2. Handle unusual logons

This topic describes how to handle unusual logons. The unusual logons include logons from disapproved locations or IP addresses, logons by using brute-force attacks or disapproved accounts, and logons at a disapproved time.

Context

For more information about the policies that are configured to identify unusual logons, see Configure policies to identify unusual logons.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   > **Note** For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Intrusion Detection**.
3. Click the **Unusual Logons** tab.
4. (Optional)Specify the filter conditions to locate unusual logons. The filter conditions include **Asset**, **Alert Type**, and **Status**.

   > **Note** If you skip this step, all unusual logons appear in the event list.

5. In the event list, view and handle unusual logons.

   Check whether an unusual logon is a false positive.

   - If the unusual logon is a false positive, click **Ignore**.
   - If the unusual logon is not a false positive, harden the security of the physical server and click **Ignore**. To harden the security of the physical server, you can specify a more complex password, fix vulnerabilities, or check configuration items.

27.1.7.4.3. Handle webshell events

This topic describes how to view and handle webshell events.

Context

The webshell detection feature uses a webshell detection engine developed by Alibaba Cloud to detect and handle webshell files. The feature handles webshell files of the PHP and JSP types.

> **Note** By default, the system checks the web directories of all protected physical servers once a day. You can specify the servers that you want the system to check. For more information, see Configure security settings for physical servers.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   > **Note** For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Intrusion Detection**.
3. In the left-side navigation pane, choose **Physical Server Security > Intrusion Detection**.
4. Click the **Webshells** tab.
5. (Optional)Configure search conditions to search for webshell events. The search conditions include **Asset** and **Status**.

   > **Note** If you skip this step, all webshell events appear in the event list.

6. In the webshell event list, view and handle webshell events.

   Check whether a webshell event is a false positive based on the webshell information.

   - If the webshell event proves to be a false positive, click **Ignore**. The system no longer reports alerts for the webshell file.
   - If the webshell event is not a false positive, manually handle the webshell file.

27.1.7.4.4. Handle server exceptions

This topic describes how to view the alerts for server exceptions and handle the server exceptions.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   > **Note** For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Intrusion Detection**.
3. Click the **Server Exceptions** tab.
4. (Optional)Configure search conditions to search for server exceptions. The conditions include **Asset**, **Event Name**, **Event Type**, **Risk Level**, and **Status**.

   > **Note** If you skip this step, all server exceptions appear in the event list.

5. In the server exception list, view and handle server exceptions.

   Find a server exception and click **View** in the **Actions** column. Then, you can view the information about the server exception and check whether the server exception is a false positive.

   - If the server exception proves to be a false positive, click **Mark as False Positive**. The system no longer reports alerts for the server exception.
   - If the server exception does not affect the security of your servers, click **Ignore** to ignore the alert.
   - If the server exception affects the security of your servers, manually handle it and click **Confirm**.
   - If the server exception is marked as a false positive but affects the security of your servers, click **Cancel Marking as False Positive**. The system reports an alert for the server exception if the server exception is detected again.

27.1.7.5. Server fingerprints

27.1.7.5.1. Configure data refresh frequencies

You can configure the frequencies at which the data of running processes, system accounts, listening ports, and software versions is collected and refreshed.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   > **Note** For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Server Fingerprints**.
3. In the upper-right corner of the Server Fingerprints page, click **Configure**.
4. In the **Configure** dialog box, configure the data refresh frequencies for the listening ports, running processes, system accounts, and software versions.
5. Click **OK**.

27.1.7.5.2. View listening ports

Apsara Stack Security regularly collects information about listening ports on a server.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   > **Note** For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Server Fingerprints**.
3. Click the **Listening Port** tab.
4. (Optional)Specify **Port Number** and **Process Name**. Then, click **Search** to search for the specified port or process.

   > **Note** If you skip this step, all listening ports appear in the port list.

5. In the port list, view **Port Number**, **Network Protocol**, and **Servers**.
6. Click a port number to view the information about the assets and processes that are associated with the port.

27.1.7.5.3. View running processes

Apsara Stack Security regularly collects the process information of a server.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   > **Note** For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Server Fingerprints**.
3. Click the **Running Process** tab.
4. (Optional)Specify **Process Name** and **User** and click **Search** to search for the specified process or user.

   > **Note** If you skip this step, all processes appear in the process list.

5. In the process list, view **Process Name** and **Servers**.

6. Click the name of a running process to view the information about the assets, paths, and running users that are associated with the process.

27.1.7.5.4. View account information

Apsara Stack Security regularly collects the account information of a server.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   > **Note** For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Server Fingerprints**.
3. Click the **Account Information** tab.
4. (Optional)Specify **Username** and click **Search** to search for the specific account.

   > **Note** If you skip this step, all accounts appear in the account list.

5. In the account list, view **Account** and **Servers**.
6. Click the name of an account to view the information about the assets and user groups that are associated with the account. You can also check whether the account has root permissions

27.1.7.5.5. View software versions

This topic describes how to view the information about the software versions of physical servers. The information facilitates the management of software assets.

Context

This topic covers the following scenarios:

- Check for software assets that are installed without authorization.
- Check for outdated versions of software assets.
- Find affected assets if vulnerabilities are detected.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   > **Note** For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Server Fingerprints**.
3. Click the **All Versions** tab.
4. (Optional)Specify **Software Name**, **Version Name**, and **Software Installation Path**. Then, click **Search** to search for specific software.

   > **Note** If you skip this step, all software appears in the software list.

5. In the software list, view the information in the **Software Name** and **Hosts** columns.
6. Click the name of the software to view the host, version, and installation directory of the software.

27.1.7.6. Log retrieval

27.1.7.6.1. Supported log sources and fields

This topic describes the log sources and fields that are supported by the log retrieval feature.

The log retrieval feature allows you to query the following types of log sources. You can click a log source link to view the fields that can be retrieved.

Log source	Description
Logon history	Log entries about successful system logons.
Logs of brute-force attacks	Log entries about failed system logons during brute-force attacks.
Process snapshot logs	Log entries about processes on a server at a specific point in time .
Logs of listening port snapshots	Log entries about listening ports on a server at a specific point in time.
Account snapshot logs	Log entries about account-based logons on a server at a specific point in time.
Process startup logs	Log entries about process startups on a server.
Network connection logs	Log entries about active connections from a server to the Internet.

Logon history

You can query the logon history by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.
IP	string	The IP address of the server.
warn_ip	string	The source IP address for the logon.
warn_port	string	The logon port.
warn_user	string	The username for the logon.
warn_type	string	The logon type.
warn_count	string	The number of logon attempts.

Logs of brute-force attacks

You can query logs about brute-force attacks by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.
IP	string	The IP address of the server.
warn_ip	string	The source IP address of the attack.

Field	Data type	Description
warn_port	string	The target port.
warn_user	string	The target username.
warn_type	string	The type.
warn_count	string	The number of brute-force attack attempts.

Process startup logs

You can query process startup logs by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.
IP	string	The IP address of the server.
pid	string	The ID of the process.
groupname	string	The user group.
ppid	string	The ID of the parent process.
uid	string	The ID of the user.
username	string	The username.
filename	string	The file name.
pfilename	string	The name of the parent process file.
cmdline	string	The command line.
filepath	string	The path of the process file.
pfilepath	string	The path of the parent process file.

Logs of listening port snapshots

You can query logs about listening port snapshots by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.
IP	string	The IP address of the server.
src_port	string	The listening port.
src_ip	string	The listening IP address.
proc_path	string	The path of the process file.
pid	string	The ID of the process.

Field	Data type	Description
proc_name	string	The name of the process.
proto	string	The protocol.

Account snapshot logs

You can query account snapshot logs by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.
IP	string	The IP address of the server.
perm	string	Indicates whether the user has root permissions.
home_dir	string	The home directory.
warn_time	string	The time at which a password expiration notification is sent.
groups	string	The group to which the user belongs.
login_ip	string	The IP address of the last logon.
last_chg	string	The time at which the password was last changed.
shell	string	The Linux shell command.
domain	string	The Windows domain.
tty	string	The logon terminal.
account_expire	string	The time at which the account expires.
passwd_expire	string	The time at which the password expires.
last_logon	string	The last logon time.
user	string	The username.
status	string	The account status. Valid values: • 0: disabled • 1: normal

Process snapshot logs

You can query process snapshot logs by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.

Field	Data type	Description
IP	string	The IP address of the server.
path	string	The path of the process file.
start_time	string	The time at which the process was started.
uid	string	The ID of the user.
cmdline	string	The command line.
pname	string	The name of the parent process.
name	string	The name of the process.
pid	string	The ID of the process.
user	string	The username.
md5	string	The MD5 hash of the process file. If the size of the process file exceeds 1 MB, the system does not calculate the MD5 hash value of the process file.

Network connection logs

You can query network connection logs by using the fields in the following table.

Field	Data type	Description
uuid	string	The ID of the client.
IP	string	The IP address of the server.
src_ip	string	The source IP address.
src_port	string	The source port.
proc_path	string	The path of the process file.
dst_port	string	The destination port.
proc_name	string	The name of the process.
dst_ip	string	The destination IP address.
status	string	The status.

27.1.7.6.2. Logical operators

The log retrieval feature supports multiple search conditions. You can add multiple logical operators to one search condition for one log source, or combine multiple search conditions for several log sources by using different logical operators. This topic describes the logical operators that are supported in log retrieval. Examples are provided to help you understand these operators.

The following table describes the logical operators that are supported in log retrieval.

Logical operators

Logical operator	Description
and	Binary operator. This operator is in the format of `query 1 and query 2`, which indicates the intersection of the query results of `query 1` and `query 2`. **Note** If no logical operators are used for multiple keywords, the default operator is AND.
or	Binary operator. This operator is in the format of `query 1 or query 2`, which indicates the union of the query results of `query 1` and `query 2`.
not	Binary operator. This operator is in the format of `query 1 not query 2`, which indicates the results that match `query 1` but do not match `query 2`. This format is equivalent to `query 1 - query 2`. **Note** If you use only `not query 1`, the log data that does not contain the query results of `query 1` is returned.

27.1.7.6.3. Query logs

This topic describes how to query and view the logs of physical servers.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   Note For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Log Retrieval**.
3. Specify search conditions.

   Note For more information about log sources, log fields, and logical operators, see Supported log sources and fields and Inference rules and logical operators.

4. Click **Search** and view the search result.
   - **Reset**: You can click **Reset** to clear the search condition configuration.
   - **Save Search**: You can click **Save Search** to save the search condition configuration for future use.
   - **Saved Searches**: You can click **Saved Searches** to select and apply a saved search condition configuration.

27.1.7.7. Configure security settings for physical servers

This topic describes how to configure security settings for physical servers. You can enable or disable periodic trojan scans. You can also configure the working mode of the Server Guard agent for physical servers.

Procedure

1. Log on to Apsara Stack Security Center by using an Apsara Stack tenant account.

   > **Note** For more information about the Apsara Stack tenant account, see Create and grant permissions to a security administrator account.

2. In the left-side navigation pane, choose **Physical Server Security > Settings**.
3. Enable periodic trojan scans for physical servers.
   i. Click **Manage** in the Trojan Scan section.
   ii. Select the physical servers on which you want to perform periodic trojan scans from the All Servers section. Then, click the rightwards arrow.
   iii. Click **OK**.
4. On the **Protection First Mode** page, click **Manage** next to Protection Mode.

   Configure protection modes for servers.

   - **Business First Mode**: In this mode, the peak CPU utilization is less than 10%, and the peak memory usage is less than 50 MB.
   - **Protection First Mode**: In this mode, the peak CPU utilization is less than 20%, and the peak memory usage is less than 80 MB.

27.1.8. Application security

27.1.8.1. Quick start

This topic helps you get started with Web Application Firewall (WAF).

WAF uses intelligent semantic analysis algorithms to identify web attacks. WAF also integrates a learning model to enhance its analysis capabilities and meet your daily security protection requirements without relying on traditional rule libraries.

The following content describes the procedure for using WAF:

1. Customize WAF protection rules.

   WAF provides a default protection policy. You can also customize policies that suit your business requirements.

   - For more information about how to configure protection policies, see Configure protection policies.
   - For more information about how to configure custom rules, see Create a custom rule.
   - For more information about how to configure HTTP flood protection rules, see Configure an HTTP flood protection rule.

2. Add protected websites.

   WAF can protect Internet websites and virtual private cloud (VPC) websites.

   - For more information about how to add an Internet website to WAF for protection, see Add an Internet website for protection.
   - For more information about how to add a VPC website to WAF for protection, see Add a VPC website for protection.

3. Configure Domain Name System (DNS) resolution.

   For more information about how to change the DNS-resolved source IP address of a website to a virtual IP address of WAF, see Modify DNS resolution settings.

4. View WAF protection results.

- For more information about how to view the protection overview, see View protection overview.
- For more information about how to view the service access information, see View Web service access information.
- For more information about how to view the detection logs for web attacks, see View attack detection logs.
- For more information about how to view the detection logs for HTTP flood attacks, see View HTTP flood protection logs.

27.1.8.2. Detection overview

27.1.8.2.1. View protection overview

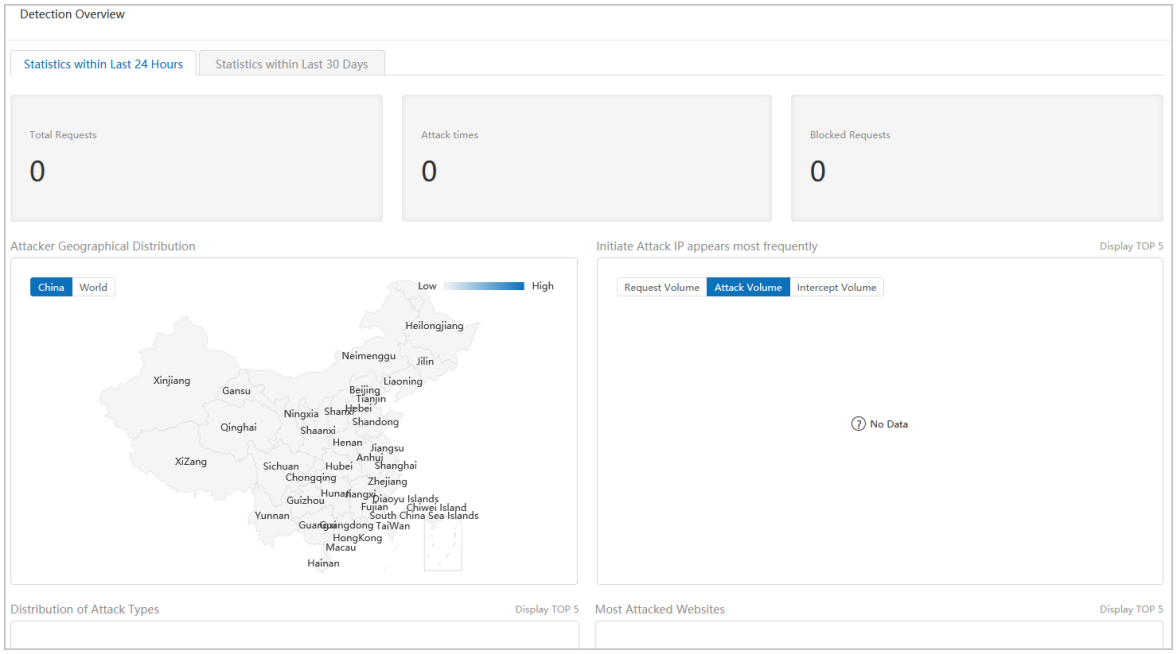
This topic describes how to view the Web Application Firewall (WAF) protection overview.

Context

The Detection Overview page displays information such as the statistics of previous attacks, the geographical distribution of attackers, the number of total requests, and the number of blocked requests. You can also view details about the attacks. This way, you can customize rules to protect your web services.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Application Security > WAF**.
3. In the left-side navigation tree, choose **Detection Overview > Detection Overview**.
4. On the **Detection Overview** page, you can view **Statistics within Last 24 Hours** and **Statistics within Last 30 Days**.



- Total Requests

  Displays the total number of requests.

- Attack times

  Displays the total number of attacks.

- Blocked Requests

  Displays the number of blocked requests.

- Geographical distribution of attackers on a map

  Displays the distribution of attackers on a map. You can select a map of China or a map of the world.

  Displays both the numbers of total requests and blocked requests.

- Distribution of the top five IP addresses that initiate attacks

  Displays the top five IP addresses that have launched the most attacks in a bar chart. The x-axis indicates the numbers of requests. The y-axis indicates the IP addresses.

- Distribution of the top five attack types

  Displays the distribution of the top five attack types and the number of attacks of each type in a pie chart.

- Top five attacked websites

  Displays the top five attacked websites and the number of attacks on each website in a bar chart.

27.1.8.2.2. View access information about web services

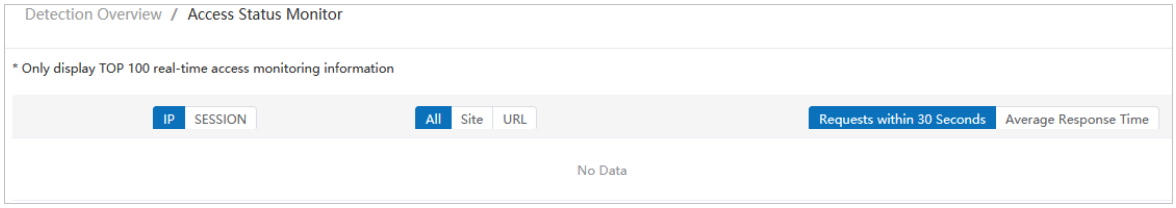
This topic describes how to view access information about web services.

Context

Web Application Firewall (WAF) monitors the access of web services. This allows security administrators to analyze the service access information and detect vulnerabilities.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Application Security > WAF**.
3. On the Detection Overview page, choose **Detection Overview > Access Status Monitor**.
4. Filter the access records to view the details.



27.1.8.3. Protection logs

27.1.8.3.1. View attack detection logs

This topic describes how to view attack detection logs.

Context

These logs allow you to analyze attacks on your web services. You can update the attack detection policies, custom rules, and fix the web service vulnerabilities based on the analysis results.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Application Security > WAF**.

3. In the left-side navigation tree, choose **Detection Logs > Attack Detection Logs**.
4. Click **Filter**, specify filter conditions, and then click **OK**.

   > **Note** If you specify multiple conditions, all the conditions must be met.

5. View the attack detection logs.

27.1.8.3.2. View HTTP flood protection logs

This topic describes how to view HTTP flood protection logs.

Context

These logs allow you to analyze HTTP flood attacks on your web services. Based on the analysis, you can update the HTTP flood protection rules and HTTP flood whitelist and fix the web service vulnerabilities.

Procedure

1.
2. In the left-side navigation pane, choose **Application Security > WAF**.
3. In the left-side navigation tree, choose **Detection Logs > HTTP Flood Detection Logs**.
4. Click **Filter**, specify filter conditions, and then click **OK**.

   > **Note** If you specify multiple conditions, all of the conditions must be met.

5. View the HTTP flood detection result.

   The blocked HTTP flood attacks, related rules, and attack time are displayed.

27.1.8.3.3. View system operation logs

This topic describes how to view system operation logs.

Procedure

1.
2. In the left-side navigation pane, choose **Application Security > WAF**.
3. In the left-side navigation tree, choose **Detection Logs > System operation log**.
4. View the system operation logs.

   The usernames, content, IP addresses, and creation time are displayed.

27.1.8.3.4. View access logs

This topic describes how to view access logs.

Procedure

1.
2. In the left-side navigation pane, choose **Application Security > WAF**.
3. On the Detection Overview page, choose **Detection Logs > Access Log**.
4. Click **Filter**, specify filter conditions, and then click **OK**.

> **Note** If you specify multiple conditions, they use the AND logic. The system returns the required access logs only when all the conditions are met.

5. View the access logs.

   The requested IP addresses, destination IP addresses, source IP addresses, methods, and response codes are displayed.

27.1.8.4. Protection configuration

27.1.8.4.1. Configure protection policies

This topic describes how to configure Web Application Firewall (WAF) protection policies.

Context

WAF provides a default protection policy. You can also customize protection policies to suit your business requirements.

Procedure

1.
2. In the left-side navigation pane, choose **Application Security > WAF > Protection Configuration > Website Protection Policies**.
3. Click **Add protection policy**. In the panel that appears, specify **Policy name** and click **Confirm**.
4. In the **Operation** column of the new protection policy, click the icon to view details.

Parameter	Description
Decode	Select algorithms that you want to use to decode the requests.
Attack Detection Modules	Specify the types of attacks that you want to detect and the risk levels of attacks that you want to block.
Block Options	Specify the HTTP status code and image that you want WAF to return when it blocks an attack.
HTTP Response Processing	Specify **Enable HTTP response processing** and **Response Detection Max Body Size**.
HTTP Request Body Detection	Specify **Response Detection Max Body Size**.
Detection Timeout	Specify **Enable Detection Timeout** and **Timeout Threshold**.

For example, perform the following steps to configure modules in the **Attack Detection Modules** section:

i. Move the pointer over a specific module in the **Attack Detection Modules** section. For this example, move the pointer over **SQL Injection Detection Module** and click the **modify** icon.

ii. In the **SQL Injection Detection Module** dialog box, configure the following parameters.

Parameter	Description
Enabled	Indicates whether to enable the detection module.
Blocking Threshold	Valid values: **NotForbid**, **Only ForbidHigh Risk**, **ForbidMedium or High Risk**, and **Forbid All**.
Record Threshold	Valid values: **Notrecord**, **Onlyrecord High Risk**, **recordMedium or High Risk**, and **record All**.
Detect Non-Injected SQL	Indicates whether to enable detection for NoSQL injection vulnerabilities.

iii. Click **OK**.

5. Manage the protection policy.

   To delete a protection policy, click the protection policy first. Then, in the upper-right corner, choose **More > Delete Selected Protection Policies**. In the dialog box that appears, click **OK**.

   > **Note** You cannot delete the default protection policy.

27.1.8.4.2. Create a custom rule

This topic describes how to create a custom rule for Web Application Firewall (WAF).

Context

You can create custom rules to meet the different requirements for intrusion detection. You can create, edit, or delete custom rules in the WAF console as an administrator. You can use custom rules to filter out requests that meet specific conditions.

Multiple custom rules use the **OR** logic. If two custom rules use the same conditions but trigger different actions such as blocking traffic or allowing traffic, the system runs the first rule.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Application Security > WAF > Protection Configuration > Customized Rules**.
3. In the upper-right corner, click **Add Rule**. In the **Add Customized Rule** panel, configure the parameters.

   Parameters for the creation of a custom rule

Parameter	Description
Type	The operating mode of the rule. Valid values: **Block**, **Allow**, **Monitor**, and **Detection module control**. ○ **Block**: If an HTTP request meets the conditions of the rule, the HTTP request is blocked. ○ **Allow**: If an HTTP request meets the conditions of the rule, the HTTP request is allowed. ○ **Monitor**: If an HTTP request meets the conditions of the rule, the HTTP request is recorded and allowed. ○ **Detection module control**

Parameter	Description
Remarks	The remarks about the rule, such as the purpose of the rule.
Risk Level	The risk level. Valid values: **No threat**, **Low Risk**, **Medium Risk**, and **High Risk**.
Matching Pattern	The conditions that trigger the rule. Click **Add Pattern** to specify more than one condition. Multiple conditions use the **AND** logic. The custom rule takes effect only if all conditions are met.
Apply to Websites	The websites that you want the rule to protect.
Log Recording Option	Whether to record a protection event in the intrusion detection logs when the rule is triggered. The default value is Enable Log Recording. After Log Recording Option is set to Enable Log Recording, all interception events are recorded in the intrusion detection logs.
Attack Type	The type of attack that you want the rule to block.
Expiration Time	The time at which the rule expires.

4. Click **Confirm**.
5. Manage the custom rule.
   - Edit the rule.

     To edit the rule, click the icon in the **Actions** column.
   - Enable the rule.

     To enable the rule, select the rule and click **Enable Selected Rules**.
   - Disable the rule.

     To disable the rule, select the rule and click **Disable Selected Rules**.
   - Delete the rule.

     To delete the rule, select the rule and click **Delete Selected Rules**.

27.1.8.4.3. Configure an HTTP flood mitigation rule

This topic describes how to configure an HTTP flood mitigation rule.

Context

An HTTP flood attack is a type of distributed denial of service (DDoS) attack that targets web applications. Attackers use proxy servers or zombies to overwhelm targeted web servers by sending a large number of HTTP requests.

Create an HTTP flood mitigation rule

1.
2. In the left-side navigation pane, choose **Application Security > WAF > Protection Configuration > HTTP Flood Detection**.
3. Click **Add Rule**. The **Add HTTP Flood Detection Rules** panel appears.
4. Configure parameters and click **Confirm**.

Parameter	Description
Rule Mode	The action on requests after the HTTP flood mitigation rule is triggered. Valid values: **Blocking Mode** and **Observe**. ○ Blocking Mode: blocks the requests that trigger the HTTP flood mitigation rule. ○ Observe: records the requests that trigger the HTTP flood mitigation rule, but does not limit the requests.
Rule Types	The type of the HTTP flood mitigation rule. Valid values: **Restrict Users by Policy** and **Restrict Known Users**. The difference between the two types is determined by whether requests of users are initiated from a specific IP address or in a specific session. ○ **Restrict Users by Policy**: limits requests that meet all the configuration items of the HTTP flood mitigation rule. Configuration items include **Restriction Trigger Threshold**, **Restricted URL Address**, **Restriction Mode**, **Restriction Time**, and **Statistical Range of Visits** in the Advanced section. ○ **Restrict Known Users**: limits requests that are initiated from specific IP addresses or in specific sessions based on the HTTP flood mitigation rule. To achieve this purpose, you must configure the IP address or session list and the restriction mode. After you configure the list, the HTTP flood mitigation rule limits requests based on the list.
Rule Name	The name of the HTTP flood mitigation rule.
Target Type	The type of sources for requests that are blocked. Valid values: **IP** and **Session**. **Note** You can set **Target Type** to **SESSION** only for a website to which the HTTP mitigation rule is applied and whose **User Identification** is set to **WAF User System**. For more information, see Add an Internet website for protection.
Restriction Trigger Threshold	If **Rule Types** is set to **Restrict Users by Policy**, you must configure the triggering conditions for the HTTP flood mitigation rule.
Restricted URL Address	If **Rule Types** is set to **Restrict Users by Policy**, you must specify the URL addresses to which requests are blocked based on the HTTP flood mitigation rule. ○ **URL Prefix** ○ **URL** ○ **Record all IP addresses**
Restricted IP List or **Restricted Session List**	If **Rule Types** is set to **Restrict Known Users**, enter the IP addresses or sessions from which you want to block requests based on the setting of **Target Type**. Specify only one IP address or session in each line.

Parameter	Description
Restricted URL Address	If **Rule Types** is set to **Restrict Known Users**, you must specify the URL addresses to which requests are blocked based on the HTTP flood mitigation rule. ○ **URL Prefix** ○ **URL** ○ **Restrict user access to all addresses**
Restriction Mode	The mode in which the HTTP flood mitigation rule limits requests. Valid values: ○ **Forbidden**: The rule blocks specific sources from accessing the specified URL address. ○ **Frequency control**: The rule limits the frequency at which specific sources access the specified URL address.
Restriction Time	The time at which the HTTP flood mitigation rule takes effect.
Statistical Range of Visits	If **Rule Type** is set to **Restrict Users by Policy**, you can specify the range of the limited requests that you want to analyze in the Advanced section. ○ Statistics Full Access Data: If you select this option, the HTTP flood mitigation rule applies to the requests that are redirected by WAF. If these requests meet the rule, they are limited. This decreases system performance. ○ Statistics TOP Access Data: If you select this option, the frequency of the requests that access the top 100 data records is limited. This option helps minimize the decrease in system performance. You can select this option when the number of accessed data records is larger than 100. Note that the top 100 data records are measured based on real-time monitoring.

Manage HTTP flood mitigation rules

1.
2.
3. In the rule list, manage existing HTTP flood mitigation rules.
   - Search for a rule.

     Click **Filter** and specify filter conditions to find a rule.
   - Enable a rule.

     Select a rule that is in the Disabled state and choose **More > Enable Selected Rules**.
   - Disable a rule.

     Select a rule that is in the Enabled state and choose **More > Disable Selected Rules**.
   - Delete a rule.

     Select a rule and choose **More > Delete Selected Rules**.

27.1.8.4.4. Configure an HTTP flood whitelist

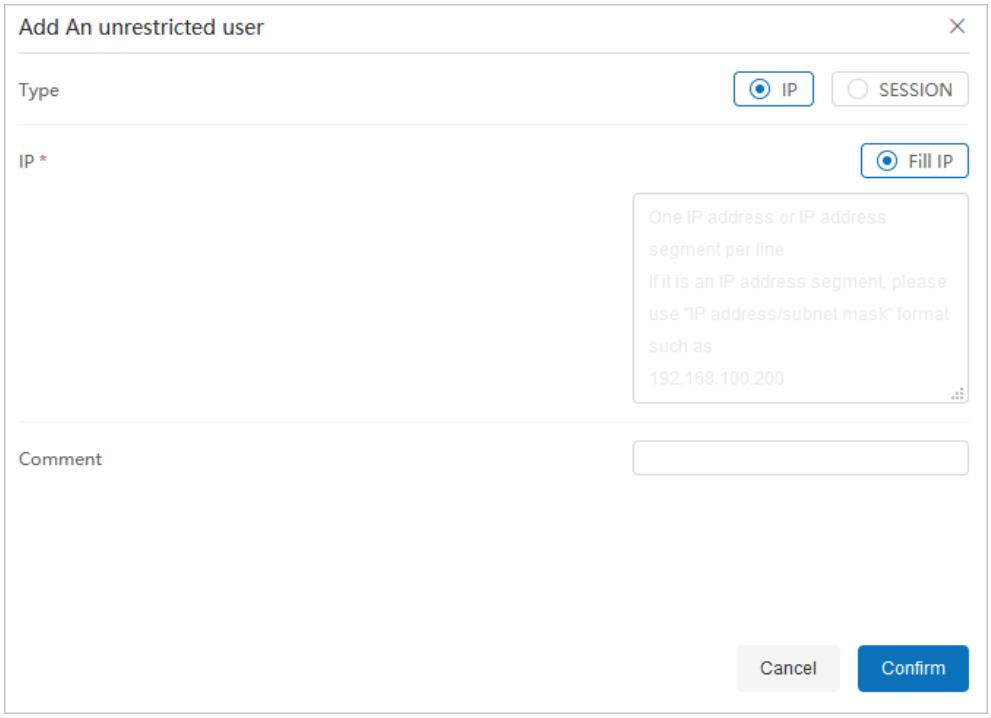
This topic describes how to configure an HTTP flood whitelist.

Context

If a request source is trusted, you can add this request source to an HTTP flood whitelist to allow the requests from this source.

Procedure

1.
2. In the left-side navigation pane, choose **Application Security > WAF > Protection Configuration > HTTP Flood Detection**. On the page that appears, click the **HTTP Flood Detection Whitelist** tab.
3. On the HTTP Flood Detection Whitelist tab, click **Add Whitelist Item** to add a request source to the whitelist. Then, click **Confirm**.



Parameter	Description
Type	Select the type of the request source. Valid values: **IP** or **Session**.
IP or **SESSION**	Specify the IP addresses or sessions based on the selected **Type**. Specify one IP address or session in each line.
Comment	Enter comments for the whitelist.

4. Manage the request sources in the whitelist.
   - Search for a request source in the whitelist.

     Click **Filter**. In the dialog box that appears, click **Add Filter Item** to find a request source.
   - Remove a request source from the whitelist.

     Select a request source and choose **More > Delete Selected Items**.

27.1.8.4.5. Manage SSL certificates

This topic describes how to upload or delete SSL certificates.

Context

After you upload an SSL certificate on the **SSL Certificate Management** page, you can select this certificate when you add an HTTPS website for protection.

> **Note** When you add an HTTPS website for protection on the **Protected Websites** page, you must select the SSL certificate that corresponds to the domain of the HTTPS website.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Application Security > WAF**.
3. On the Detection Overview page, choose **Protection Configuration > SSL Certificate Management**.
4. Upload a new certificate.
   i. Click **Upload SSL Certificate**.
   ii. In the **Add File** panel, specify **Name**.
   We recommend that you enter the domain name for easier management.
   > **Note** If your certificate and private key are in the same file, select **Include private key in certificate file**.
   iii. In the **File** section, upload the Certificate Authority (CA) certificate file and private key file.
   iv. Specify **Certificate Password**.
   v. Click **Confirm**.
5. (Optional)Delete the uploaded SSL certificate.
   You can delete expired SSL certificates.
   i. In the SSL certificate list, select the certificate that you want to delete.
   ii. Choose **More > Delete selected SSL Certificates**.
   iii. In the dialog box that appears, click **Confirm**.

27.1.8.4.6. Add Internet websites for protection

This topic describes how to add Internet websites to Web Application Firewall (WAF).

Context

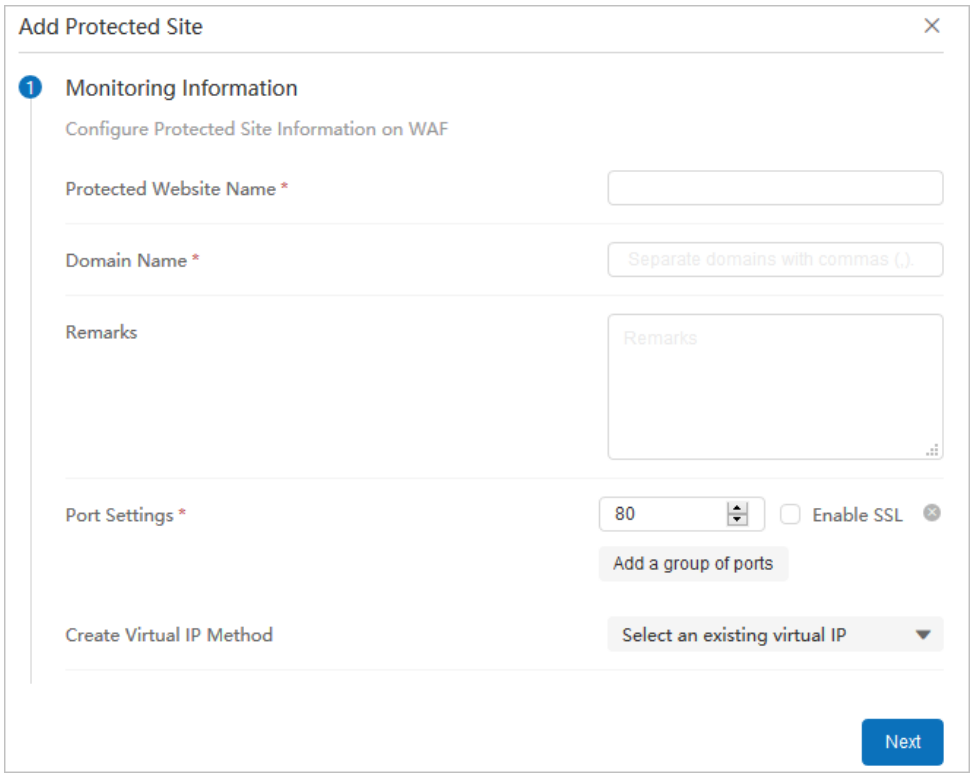
WAF can protect the following types of websites:

- Internet websites.
- Virtual Private Cloud (VPC) websites. For more information about how to add a VPC website to WAF, see Add a VPC website for protection.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Application Security > WAF**.
3. In the left-side navigation tree, choose **Protection Configuration > Protection Site Management**. On the page that appears, click the **Internet Websites** tab.
4. In the upper-right corner, click **Add a site**. The Add Protected Site panel appears.
5. In the Monitoring Information step, configure parameters and click **Next**.

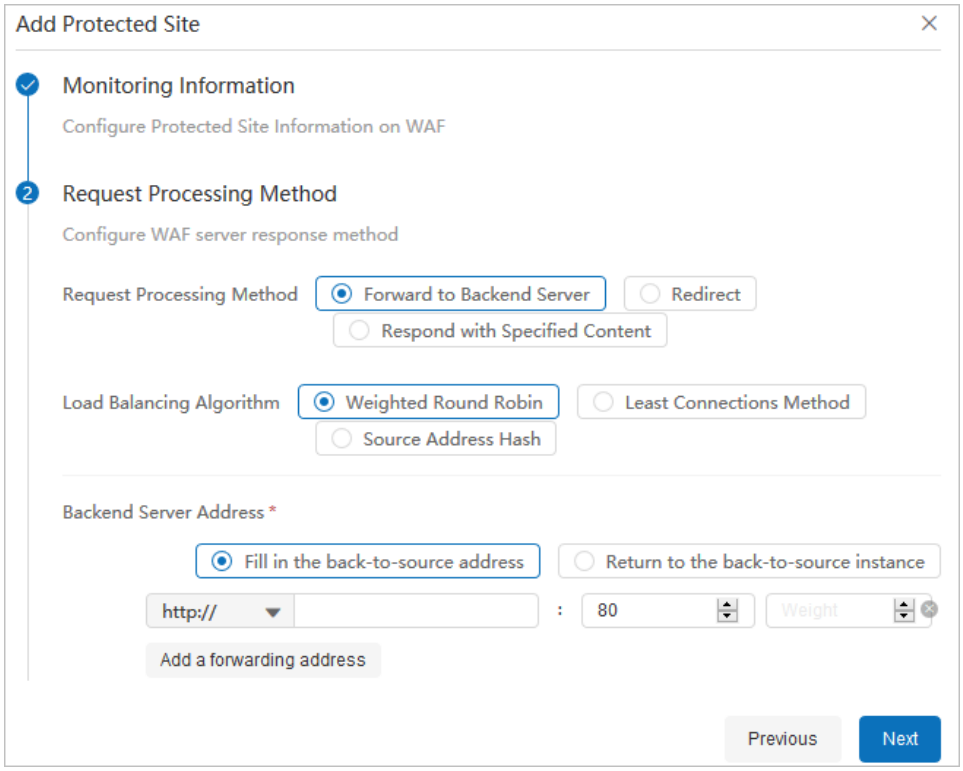
Specify the Internet website that you want WAF to protect. WAF can protect both HTTP and HTTPS websites.



Parameter	Description
Protected Website Name	The name of the website that you want WAF to protect.
Domain Name	The domain name of the website. ○ You can use an asterisk (*) as a wildcard. ○ If you specify multiple domain names, separate them with commas (,).
Port Settings	The port that WAF listens on. ○ If the website supports HTTPS requests, select **Enable SSL** and upload an HTTPS certificate. ○ If the website can be accessed over multiple ports, click **Add a group of ports** to add the required ports.

Parameter	Description
Cert Settings	The HTTPS certificate of the website. Valid values: *Upload a New Certificate* and *Choose an Existing Certificate*. ◦ *Upload a New Certificate*: If the HTTPS certificate used by the website has not been uploaded to WAF, select this option. By default, the HTTPS certificate and private key are separately uploaded. If you select **Include private key in certificate file**, upload only a file that contains both the HTTPS certificate and private key. ◦ *Choose an Existing Certificate*: If the HTTPS certificate used by the website has been uploaded to WAF, select this option. Then, select the required HTTPS certificate from the drop-down list. **Note** Specify this parameter only if you select **Enable SSL** next to the **Listening Port** field.
Name	The name of the HTTPS certificate. **Note** Specify this parameter only if you select **Enable SSL** next to **Port Settings** and set Cert Setting to Upload a New Certificate.
File	The HTTPS certificate and private key. By default, the HTTPS certificate and private key are separately uploaded. If you select **Include private key in certificate file** next to **Name**, upload only a file that contains both the HTTPS certificate and private key. **Note** Specify this parameter only if you select **Enable SSL** next to **Port Settings** and set Cert Setting to Upload a New Certificate.
Virtual IP	The IP address type and virtual IP address. **Note** You can select an IPv6 address as the virtual IP address of WAF. By default, WAF provides 10 virtual IP addresses. However, you can add more virtual IP addresses based on your business requirements. **Note** A virtual IP address is available only for the department to which the creator of the virtual IP address belongs.

6. In the Request Processing Method step, configure parameters and click **Next**.



Response mode	Parameter	Description
Forward to Backend Server	**Load Balancing Algorithm**	The algorithm for load balancing. Valid values: **Weighted Round Robin**, **Source Address Hash**, and **Least Connections Method**.
	Backend Server Address	The IP address of the origin server to which WAF forwards inbound traffic. Valid values: **Fill in the back-to-source address** and **Return to the back-to-source instance**. ○ **Fill in the back-to-source address**: Enter the address of the origin server. If you enter multiple addresses, load balancing is performed based on the specified **load balancing algorithm**. ○ **Return to the back-to-source instance**: Enter the address of a specific ECS or SLB instance. If you enter multiple addresses, load balancing is performed based on the specified **load balancing algorithm**.
	X-Forwarded-For	The passthrough mode of the actual source IP address. The X-Forwarded-For (XFF) HTTP header is used to identify the actual source IP address of an HTTP request. The X-Forwarded-For (XFF) HTTP header is used for request forwarding services, such as HTTP proxy and load balancing.

Response mode	Parameter	Description
Redirect	**Response Status Code**	The HTTP status code that WAF returns when it forwards inbound traffic to a specific URL. Valid values: **301**, **302**, and **307**. ○ **301**: The requested page has been permanently moved to another URL. ○ **302**: The requested page has been temporarily moved to another URL. The requester must continue to use the original URL for future requests. ○ **307**: The requested page has been temporarily moved to another URL. The requester must continue to use the original URL for future requests.
	Redirect address	The required URL of redirection.
Respond with Specified Content	**Response Status Code**	The HTTP status code that WAF returns when it forwards a response with specified content. Valid values: **200**, **404**, and **503**.
	Response	The content of the response. For example, you can upload an image for the **Response** parameter. When a user visits the website, WAF returns the uploaded image.

7. In the Protection Policy step, configure parameters and click Next. Then, go to the **Finish** step.

> **Note** You can configure a protection policy only if **Request Processing Method** is set to **Forward to Backend Server**.

Parameter	Description
Protection Policy	Select a WAF protection policy. For more information, see Configure protection policies.
User Identification	Specify whether to enable the user identification feature. **Note** If you enabled HTTP flood mitigation for the protected website and set **Target Type** to **Session** when you configured request limits, you must set **User Identification** to **WAF User System**.

27.1.8.4.7. Add VPC websites for protection

This topic describes how to add Virtual Private Cloud (VPC) websites to Web Application Firewall (WAF) for protection.

Context

WAF can protect the following types of websites:

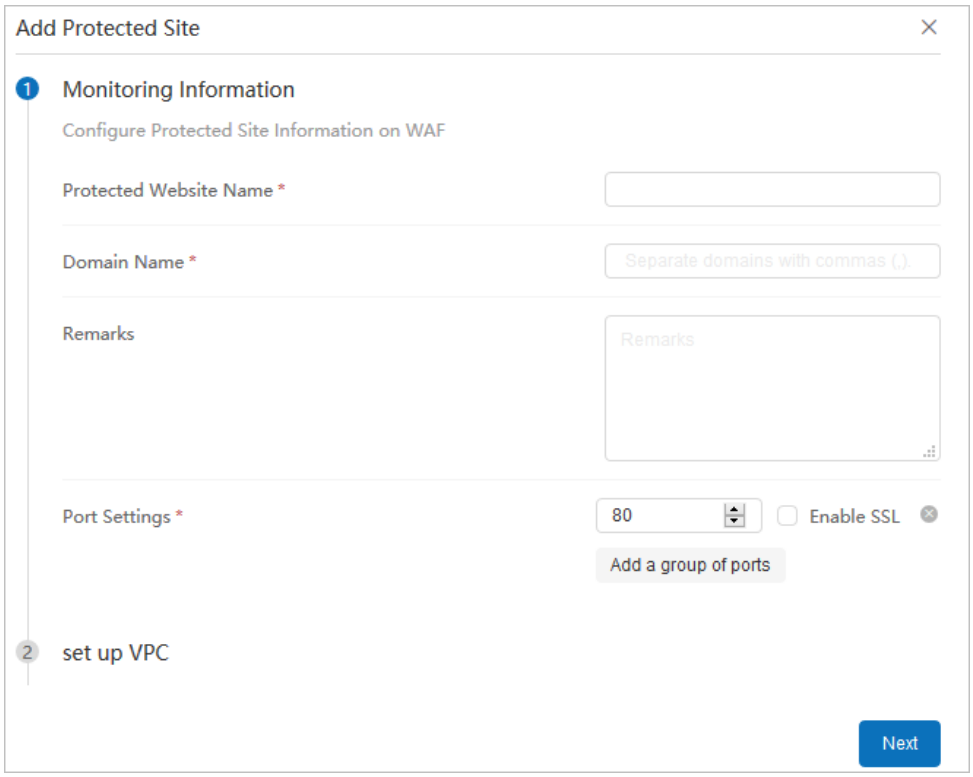
- Internet websites. For more information about how to add an Internet website for protection, see Add an Internet website for protection.

- VPC websites.

Procedure

1.
2. In the left-side navigation pane, choose **Application Security > WAF > Protection Configuration > Protection Site Management**. On the page that appears, click the **VPC Websites** tab.
3. In the upper-right corner, click **Add a site**. The **Add Protected Site** panel appears.
4. In the Monitoring Information step, configure parameters and click **Next**.

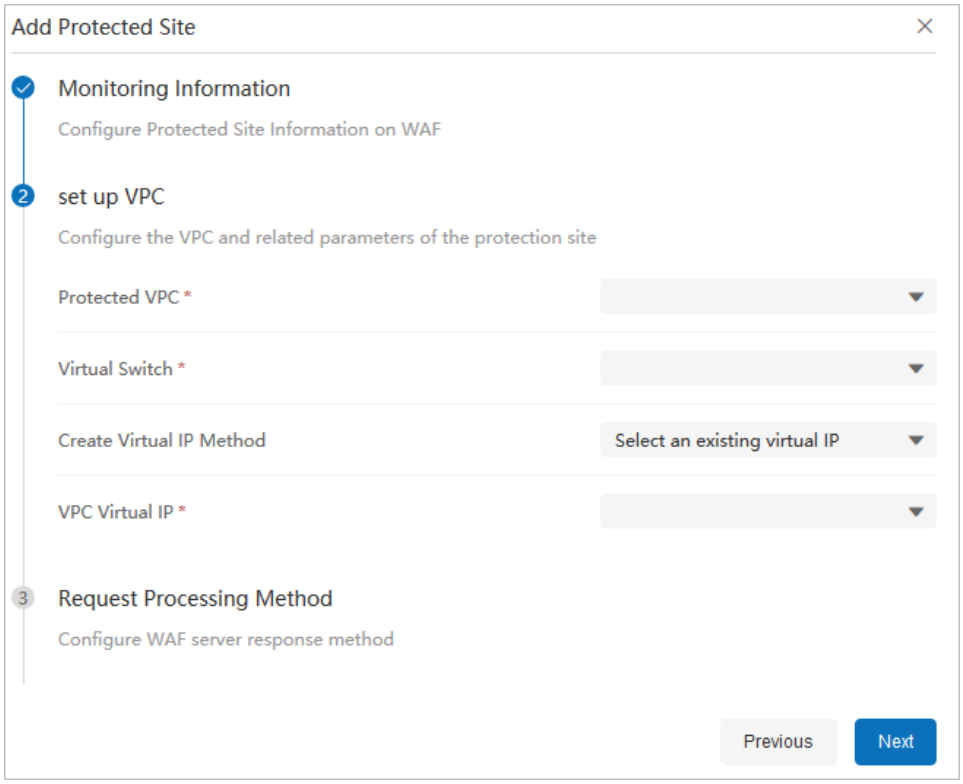
   Specify the VPC website that you want WAF to protect. WAF can protect both HTTP and HTTPS websites.



Parameter	Description
Protected Website Name	The name of the website that you want WAF to protect.
Domain Name	The domain name of the website. ○ You can use an asterisk (*) as a wildcard. ○ If you specify multiple domain names, separate them with commas (,).
Port Settings	The port that WAF listens on. ○ If the website supports HTTPS requests, select **Enable SSL** and upload an HTTPS certificate. ○ If the website can be accessed over multiple ports, click **Add a group of ports** to add the specific ports.

Parameter	Description
Cert Settings	The HTTPS certificate of the website. Valid values: **Upload a New Certificate** and **Choose an Existing Certificate**. **Note** Specify this parameter only if you select **Enable SSL** next to **Port Settings**. ○ **Upload a New Certificate**: If the HTTPS certificate used by the website has not been uploaded to WAF, select this option. By default, the HTTPS certificate and private key are separately uploaded. If you select **Include private key in certificate file**, upload only a file that contains both the HTTPS certificate and private key. ○ **Choose an Existing Certificate**: If the HTTPS certificate used by the website is uploaded to WAF, select this option. Then, select the specific HTTPS certificate from the drop-down list.
Name	The name of the HTTPS certificate. **Note** Specify this parameter only if you select **Enable SSL** next to **Port Settings** and set Cert Setting to Upload a New Certificate.
File	The HTTPS certificate and private key to upload. By default, the HTTPS certificate and private key are separately uploaded. If you select **Include private key in certificate file** next to **Name**, upload only a file that contains both the HTTPS certificate and private key. **Note** Specify this parameter only if you select **Enable SSL** next to **Port Settings** and set Cert Setting to Upload a New Certificate.

5. In the set up VPC step, configure parameters and click **Next**.



Parameter	Description
Protected VPC	The VPC to which the website belongs.
Virtual Switch	The vSwitch associated with the specified VPC.
Create Virtual IP Method	The method to create a virtual IP address. Valid values: **Select an existing virtual IP** and **Create virtual IP**.
VPC Virtual IP	◦ If you set **Create Virtual IP Method** to **Select an existing virtual IP**, select an existing virtual IP address from the **VPC Virtual IP** drop-down list. ◦ If you set **Create Virtual IP Method** to **Create virtual IP**, click **Click to Create Vip** next to **VPC Virtual IP** to generate a virtual IP address.

6. In the Request Processing Method step, configure parameters and click **Next**.

Response mode	Parameter	Description
	Load Balancing Algorithm	The algorithm for load balancing. Valid values: **Weighted Round Robin**, **Source Address Hash**, and **Least Connections Method**.

Response mode	Parameter	Description
Forward to Backend Server	**Backend Server Address**	The IP address of the origin server to which WAF forwards inbound traffic. Valid values: **Fill in the back-to-source address** and **Return to the back-to-source instance**. ○ **Fill in the back-to-source address**: Enter the address of the origin server. If you enter multiple addresses, load balancing is performed based on the specified **load balancing algorithm**. ○ **Return to the back-to-source instance**: Enter the address of a specific Elastic Compute Service (ECS) or Server Load Balancer (SLB) instance. If you enter multiple addresses, load balancing is performed based on the specified **load balancing algorithm**.
	X-Forwarded-For	The passthrough mode of the actual source IP address. The X-Forwarded-For (XFF) header is used to identify the actual source IP address of an HTTP client. The header is used for request forwarding services, such as HTTP proxy and load balancing.
Redirect	**Response Status Code**	The HTTP status code that WAF returns when it forwards inbound traffic to a specified address. Valid values: **301**, **302**, and **307**. ○ **301**: The requested page is permanently moved to another URL. ○ **302**: The requested page is temporarily moved to another URL. The requester must continue to use the original URL for future requests. ○ **307**: The requested page is temporarily moved to another URL. The requester must continue to use the original URL for future requests.
	Redirect address	The required URL for redirection.
Respond with Specified Content	**Response Status Code**	The HTTP status code that WAF returns when it returns specified content. Valid values: **200**, **404**, and **503**.
	Response	The content to return. For example, you can upload an image for the **Response** parameter. If a user visits the website, WAF returns the uploaded image.

7. In the Protection Policy step, configure parameters and click Next. Then, go to the **Finish** step.

Parameter	Description
Protection Policy	Select a WAF protection policy. For more information, see Configure protection policies.
User Identification	Specify whether to enable the user identification feature.

27.1.8.4.8. Verify the configurations of a website on your on-premises server

This topic describes how to verify the configurations of a website on your on-premises server.

Context

Before you use Web Application Firewall (WAF) to scrub traffic destined for a website, we recommend that you verify the configurations of the website on your on-premises server. After you add the virtual IP address and the domain of a website to the hosts file on your on-premises server, the request to access the domain from a local browser passes through WAF first.

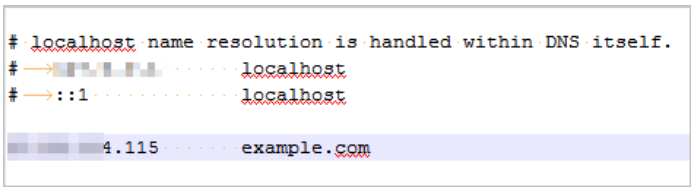
Procedure

1.
2. Add the virtual IP address and domain name to the *hosts* file on your on-premises server.

   If your computer runs Windows 7, the hosts file is stored in the following path: *C:\Windows\System32\drivers\etc\hosts*.

   i. Open the hosts file by using a text editor, such as Notepad.

   ii. Add the following content to the end of the file: `<The virtual IP address that is assigned by WAF><Protected domain name>` .

   > **Note** The IP address preceding the domain name is the virtual IP address that is assigned by WAF.

3. Ping the protected domain name from your on-premises server.

   The returned IP address must be the virtual IP address that is assigned by WAF in the hosts file. If the returned IP address is still the IP address of the origin server, refresh the local Domain Name System (DNS) cache.

4. Enter the domain name in the address bar of your browser and press Enter.

   If the access configurations on WAF are correct, you can visit the website.

5. Verify the protection capability of WAF.

   Simulate a web attack request and check whether WAF blocks the request.

   For example, add `/?alert(xss)` after the URL. If you try to visit `www.example.com/?alert(xss)` , WAF is expected to block the request.

27.1.8.4.9. Modify DNS resolution settings

This topic describes how to modify the Domain Name System (DNS) resolution settings to connect your website to Web Application Firewall (WAF).

Context

Before you modify the DNS resolution settings, verify the configurations of the website on your computer and make sure that the configurations are correct.

The domain name of a protected website may not be resolved by a DNS provider. For example, a website may use a Server Load Balancer (SLB) instance to connect to the Internet. To protect the website by using WAF, perform the following steps. Specify the virtual IP address that WAF assigns to your website as the back-to-origin IP address of the SLB instance.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Application Security > WAF**.
3. On the Detection Overview page, choose **Protection Configuration > Protection Site Management**.
4. Find the required website and click the icon in the **Operation** column.
5. On the **Basic Information** tab, obtain the virtual IP address of the protected website.
6. Log on to the console of the DNS provider and find the DNS resolution settings for the domain name of the website. Then, change the IP address in the A record to the virtual IP address of the protected website.

> **Note** We recommend that you set the TTL to 600 seconds in DNS resolution settings. The greater the TTL is, the longer it takes to synchronize and update DNS records.

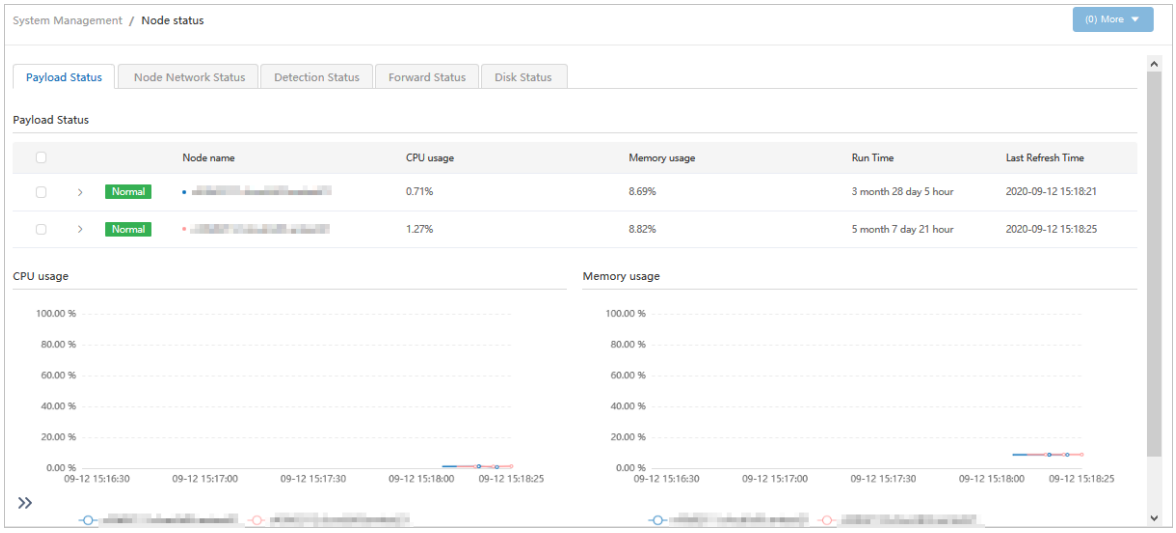
27.1.8.5. System management

27.1.8.5.1. View the load status of nodes

This topic describes how to view the load status of Web Application Firewall (WAF) nodes. The status information includes CPU utilization and memory usage. You can use the query results to identify faults or check whether scale-out is required.

Procedure

1.
2. In the left-side navigation pane, choose **Application Security > WAF**. On the Detection Overview page, choose **System Info > Node status**.
3. On the **Payload Status** tab, view the load status of WAF nodes.



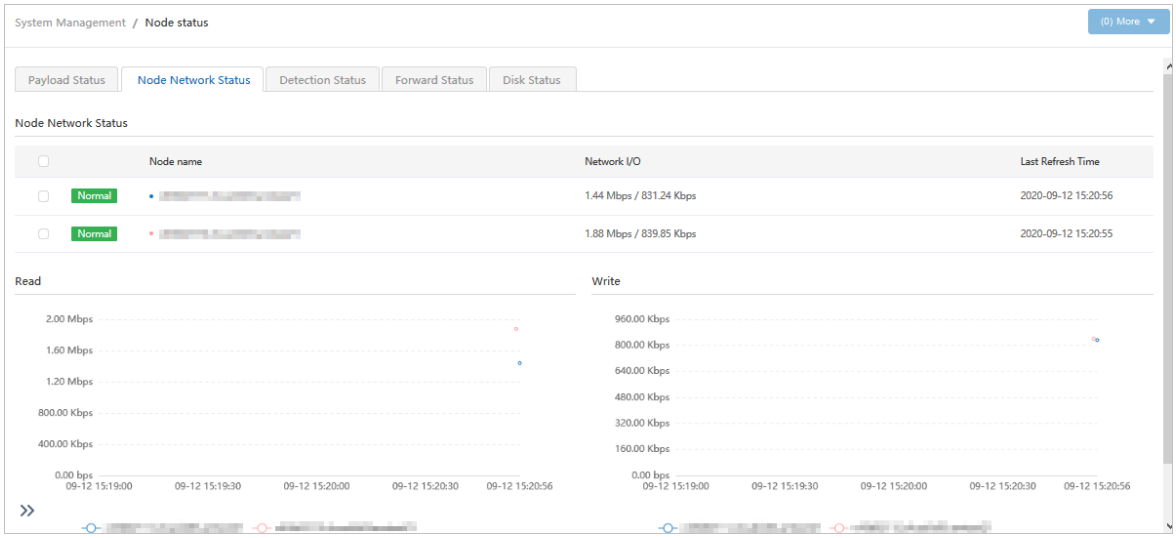
In the Payload Status section, you can view the CPU utilization and memory usage of each WAF node. In the CPU usage and Memory usage sections, you can view changes in the CPU utilization and memory usage over a specific period of time.

27.1.8.5.2. View the network status of nodes

This topic describes how to view the network status of WAF nodes, such as the network I/O, traffic detection status, and traffic forwarding status.

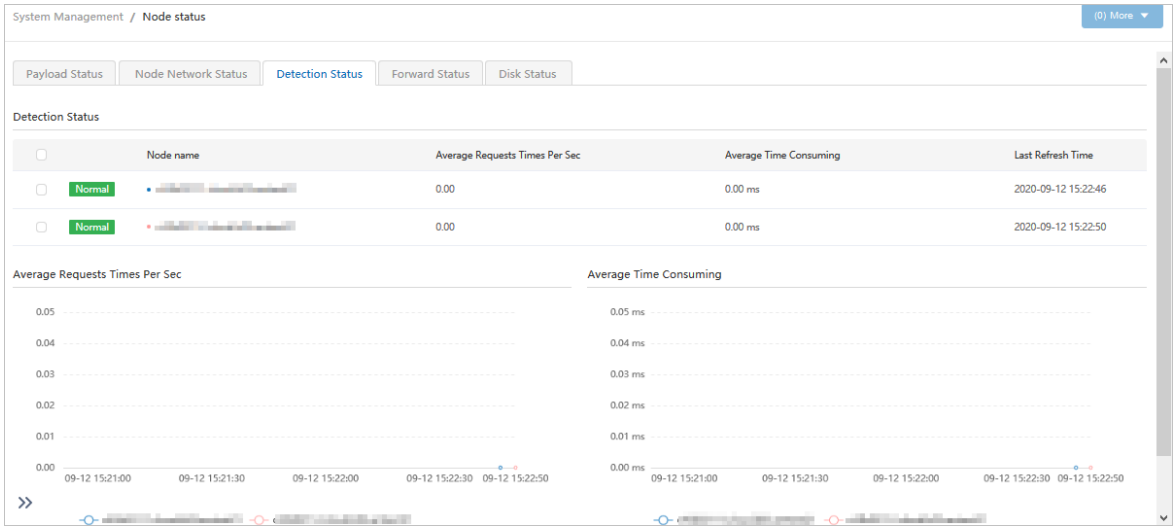
Node network status

1.
2. In the left-side navigation pane, choose **Application Security > WAF > System Management > Node status**.
3. Click the **Node Network Status** tab.
4. View the network I/O of WAF nodes.



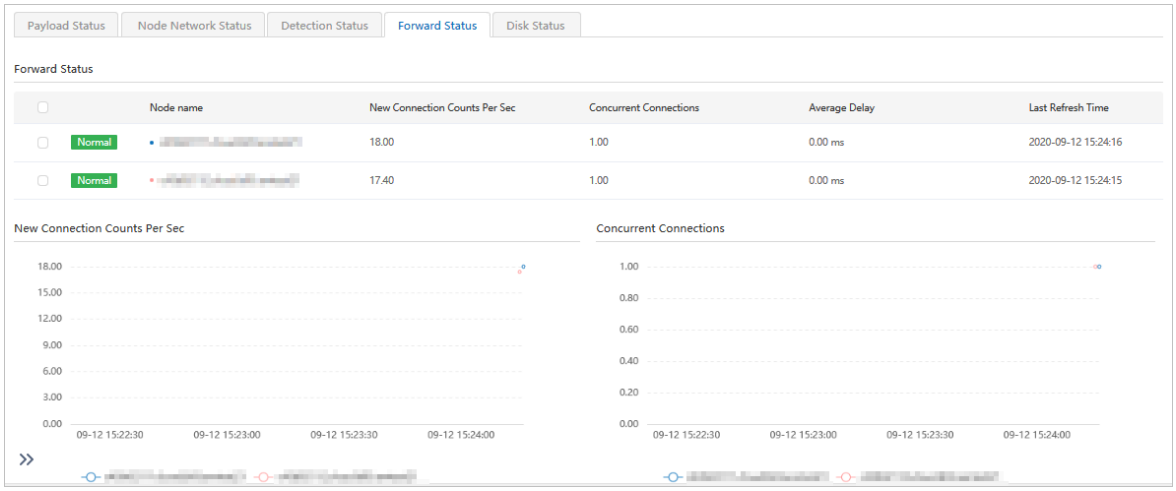
Traffic detection status

1.
2. In the left-side navigation pane, choose **Application Security > WAF > System Management > Node status**.
3. Click the **Detection Status** tab.
4. View the traffic detection status of WAF nodes.



Traffic forwarding status

1.
2. In the left-side navigation pane, choose **Application Security > WAF > System Management > Node status**.
3. Click the **Forward Status** tab.
4. View the traffic forwarding status of WAF nodes.

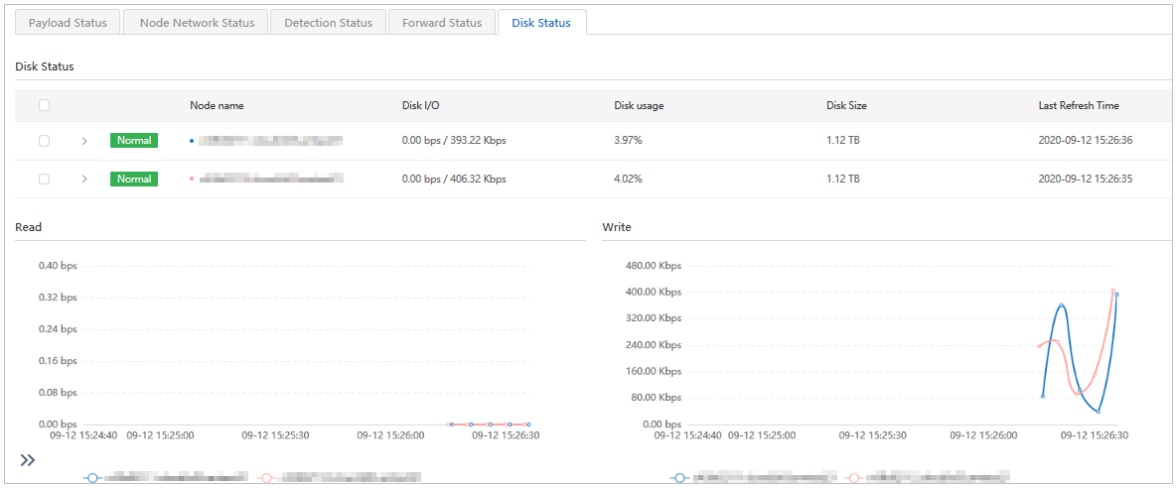


27.1.8.5.3. View the disk status of nodes

This topic describes how to view the disk status of Web Application Firewall (WAF) nodes. You can identify faults based on the status and check whether scaling is required.

Procedure

1.
2. In the left-side navigation pane, choose **Application Security > WAF > System Management > Node status**.
3. Click the **Disk Status** tab to view the disk status of WAF nodes.



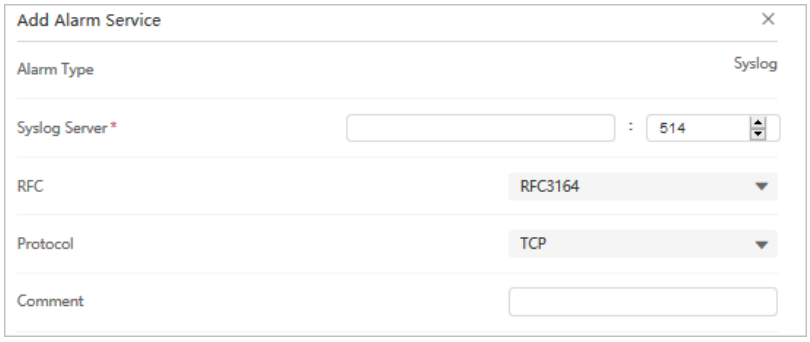
In the Disk Status section, you can view the disk I/O and disk usage of nodes. In the **Read** and **Write** sections, you can view the disk read and write changes over a period.

27.1.8.5.4. Configure alerts

This topic describes how to add a syslog server to Web Application Firewall (WAF). After the syslog server is added, WAF alert logs are pushed to the syslog server over the syslog protocol.

Procedure

1.
2. In the left-side navigation pane, choose **Application Security > WAF**. On the Detection Overview page, choose **System Management** > **Syslog Configuration**.
3. On the **Alarm Service Configuration** tab, click **Add alarm service**.
4. In the **Add Alarm Service** panel, configure parameters.



Parameter	Description
Syslog Server	The IP address and port number of the syslog server.
RFC	The Request for Comments (RFC) document that defines the Syslog protocol. Valid values: **RFC3164** and **RFC5424**.
Protocol	The transmission protocol. Valid values: **TCP** and **UDP**.
Comment	The description of the syslog server. This information facilitates subsequent identification and management.
Security	The module for which syslog logs are sent.

5. Click **Confirm**. The newly added syslog server appears in the syslog list.
6. Find the newly added syslog server and click the icon in the **Operation** column to test whether alerts are sent.
   - If a message appears, indicating that the alert test is successful, the syslog server is added.
   - If an error message appears, WAF cannot connect to the syslog server.

27.1.8.5.5. Configure alert thresholds

This topic describes how to configure alert thresholds.

Procedure

1.
2. In the left-side navigation pane, choose **Application Security > WAF**. On the Detection Overview page, choose **System Settings** > **Syslog Configuration**.
3. Click the **Alarm ThresholdConfiguration** tab and click the icon next to the thresholds that you want to modify.

4. In the Edit Queries per second panel, specify the thresholds.

Threshold	Description
Concurrent Connections	If a large number of concurrent connections exist, no alerts are sent.
Number of new connections	If a large number of new connections exist, no alerts are sent.
CPU usage is too high	If the CPU utilization exceeds this threshold in a period of time, alerts are sent.
Memory usage is too high	If the memory usage exceeds this threshold in a period of time, alerts are sent.
Disk usage is too high	If the disk usage exceeds this threshold, alerts are sent.

5. Click **Ok**.

27.1.9. Security Operations Center (SOC)

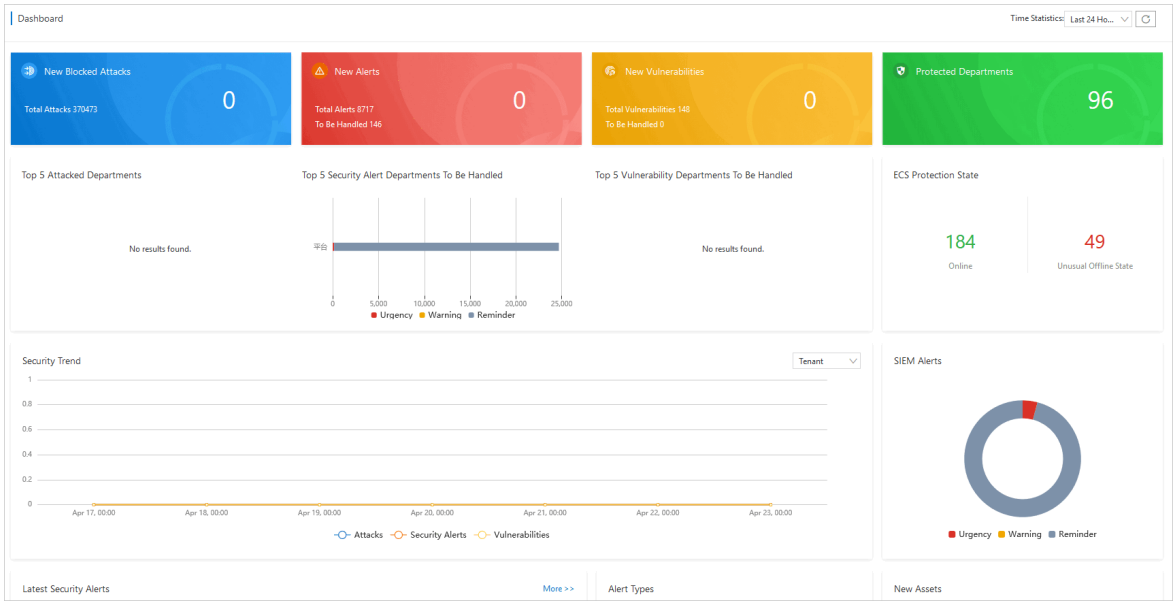
27.1.9.1. View the dashboard

This topic describes how to view the overall security information about the Apsara Stack network environment.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Overview**.
3. In the upper-right corner of the **Dashboard** page, select a time range from the **Time Statistics** drop-down list.

   Valid values: *Last 24 Hours*, *Last 7 Days*, and *Last 30 Days*.
4. View the overall security information.



The Dashboard page displays the following information:

- **New Blocked Attacks**, **New Alerts**, **New Vulnerabilities**, and **Protected Departments**
- **Top 5 Attacked Departments**, **Top 5 Security Alert Departments To Be Handled**, and **Top 5 Vulnerability Departments To Be Handled**

- **Security Trend** based on the tenant and the platform
- **Latest Security Alerts** and **Alert Types**
- **Latest Attacks** and **Attack Types**
- **ECS Protection State**, **SIEM Alerts**, **New Assets**, and **Protected Assets**

27.1.9.2. Security Monitoring

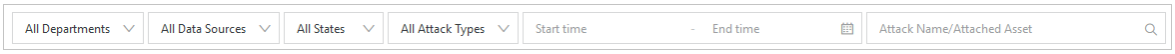
27.1.9.2.1. View the security monitoring data of tenants

This topic describes how to view the security monitoring data of tenants on the Attack Protections, Security Alerts, and Vulnerabilities tabs.

Attack Protections

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Security Monitoring > Tenant Security Monitoring**. On the page that appears, click the **Attack Protections** tab.
3. Specify search conditions.

   > **Note** If you want to view all attack events, skip this step.

Search condition	Description
Department	The department to which the assets affected by the attack belong.
Data source	The data source.
State	The attack status.
Attack type	The attack type.
Start time and end time	The time range to query.
Attack name or asset keyword	The keyword of the attack name or the affected asset.

4. View the details in the attack list.
5. Click the icons in the upper-left corner to refresh or export the list.

   The following list describes the operations:

   - Click the icon to refresh the attack list.
   - Click the icon to export the attack list.

6. In the Actions column of an attack event, you can block requests from a specific IP address, create a tag for the event, and view logs and details of the event.

   The following list describes the operations:

   - Block requests from a specific IP address: Click **Block IP Addresses**. In the **Block IP Addresses** dialog box, configure parameters to block requests from a specific IP address. For more information, see Block IP Addresses.

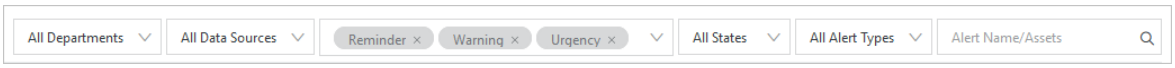
Click **View Blocked IPs** in the upper-right corner to view the blocked risk items.

- Create a tag: Click **Tag**. In the **Customize Tag** dialog box, create a tag for the attack event and click **OK**.
- View logs: Click **View Log**. On the **Cloud Tenant Logs** tab of the **Log Audit** page, view the logs of the tenant.
- View details: Click **Details** to view the details of the attack event.

Security Alerts

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Security Monitoring > Tenant Security Monitoring**. On the page that appears, click the **Security Alerts** tab.
3. (Optional)Specify search conditions.

   > **Note** If you want to view all security alerts, skip this step.

Search condition	Description
Department	The department to which the assets associated with the security alerts belong.
Source	The data source.
Level	The alert level. You can select one or more alert levels. Valid values: ○ Urgency ○ Warning ○ Reminder
Alert state	The alert status.
Type	The alert type. This can be set to **All Alert Types** or a specific alert type.
Start time and end time	The time range to query.
Alert name or asset keyword	The keyword of the alert name or the affected asset.

4. View details in the security alert list.
5. Click the icons in the upper-left corner to refresh or export the list.

   The following list describes the operations:

   - Click the icon to refresh the security alert list.
   - Click the icon to export the security alert list.

Vulnerabilities

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Security Monitoring > Tenant Security Monitoring**. On the page that appears, click the **Vulnerabilities** tab.

3. Click the **Vulnerabilities** or **Server Configurations** tab.
   - **Vulnerabilities**: provides vulnerability information.
   - **Server Configurations**: lists risks on server configuration.
4. Specify search conditions.

   > **Note** If you want to view all vulnerabilities or server configuration risks, skip this step.

Search condition	Description
Department	The department to which the assets affected by the vulnerability or server configuration risk belong.
Level	The vulnerability level or server configuration risk level.
Type	The vulnerability type or server configuration risk type.
State	The status of the vulnerability or the server configuration risk.
Start time and end time	The time range to query.
Vulnerability name, asset, or CVE ID keyword	The keyword of the name of the vulnerability or server configuration risk, CVE ID, or affected assets.

5. Click the icons in the upper-left corner to refresh or export a list of vulnerabilities or server configuration risks.

   The following list describes the operations:

   - Click the icon to refresh the list of vulnerabilities or server configuration risks.
   - Click the icon to export the list of vulnerabilities or server configuration risks.

27.1.9.2.2. View security monitoring data of platforms

This topic describes how to view the security monitoring data of platforms on the Attack Protections, Security Alerts, and Vulnerabilities tabs.

Attack Protections

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Security Monitoring > Platform Security Monitoring**. On the page that appears, click the **Attack Protections** tab.
3. Specify search conditions to search for attack protection events.

   > **Note** If you want to view all events, skip this step.

Search condition	Description
Data source	The data source.

Search condition	Description
Status	The attack status.
Attack Type	The attack type.
Start time and end time	The time range to query.
Attack name or asset keyword	The attack name or the keywords of the affected asset.

4. View the details in the attack list.
5. Click the buttons in the upper-left corner to refresh or export the list.

   The following list describes how to perform the operations:

   - Click the icon to refresh the attack event list.
   - Click the icon to export the attack event list.
6. In the Actions column of an attack event, you can block requests from a specific IP address, create a tag for the event, and view logs and details of the event.

   The following list describes how to perform the preceding operations:

   - Block requests from a specific IP address: Click **Block IP Addresses**. In the **Block IP Addresses** dialog box, configure parameters to block requests from a specific IP address. For more information, see Block IP Addresses.

     Click **View Blocked IPs** in the upper-right corner to view the blocked risk items.
   - Create a tag: Click **Tag**. In the **Customize Tag** dialog box, create a tag for the attack event and click **OK**.
   - View logs: Click **View Log**. On the **Cloud Platform Logs** tab of the **Global Log Audit** page, view the logs of the platform.
   - View details: Click **Details** to view the details of the attack event.

Security Alerts

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Security Monitoring > Platform Security Monitoring**. On the page that appears, click the **Security Alerts** tab.
3. (Optional)Specify search conditions.

   > **Note** If you want to view all security alerts, skip this step.

Search condition	Description
Data source	The data source.
Level	The alert level. You can select one or more alert levels. Valid values: ○ Urgency ○ Warning ○ Reminder
Alert state	The alert status.

Search condition	Description
Alert type	The alert type, which can be set to **All Alert Types** or a specific alert type.
Start time and end time	The time range to query.
Alert name or asset keyword	The alert name or the keywords of affected assets.

4. View details in the security alert list.
5. Click the buttons in the upper-left corner to refresh or export the list.

The following list describes how to perform the operations:

- Click the icon to refresh the security alert list.
- Click the icon to export the security alert list.

Vulnerabilities

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Security Monitoring > Platform Security Monitoring**. On the page that appears, click the **Vulnerabilities** tab and view baseline risks in the **Platform Baseline** section.
3. Specify search conditions.

> **Note** If you want to view all baseline risks of the platform, skip this step.

Search condition	Description
Level	The risk level.
Type	The baseline risk type.
State	The processing status.
Start time and end time	The time range to query.
Risk name or asset name	The risk name or the keywords of the affected asset.

4. Click the buttons in the upper-left corner to refresh or export the list of platform baseline risks.

   The following list describes the operations:

   - Click the icon to refresh the list of platform baseline risks.
   - Click the icon to export the list of platform baseline risks.

27.1.9.2.3. View the global traffic

This topic describes how to view the global traffic, including the average traffic, peak traffic, overall traffic trends, traffic of tenants, and traffic of platforms.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Security Monitoring > Global Traffic Analysis**.
3. On the **Global Traffic Analysis** page, view the global traffic.

   You can view the following information on this page:

   - View the average traffic and peak traffic
     a. In the upper-right corner of the **Global Traffic Analysis** page, select a time range and traffic direction.

        Valid values of time ranges: **Last 6 Hours**, **Last 24 Hours**, and **Last 7 Days**.

        Valid values of traffic directions: **Inbound** and **Outbound**.

     b. In the upper-left corner of the Global Traffic Analysis page, view the average and peak traffic of the specified traffic direction within the specified time range.
   - View traffic trends
     a. In the upper-right corner of the **Global Traffic Analysis** page, select a traffic type.
     b. View the overall traffic trends of each traffic type within the specified time range.
   - View the traffic of tenants on the **Tenant Traffic** tab
   - View the traffic of platforms on the **Platform Traffic** tab

27.1.9.3. Asset Management

27.1.9.3.1. View tenant assets

This topic describes how to view the assets of users. The assets include Elastic Compute Service (ECS) instances, ApsaraDB RDS instances, Object Storage Service (OSS) buckets, Server Load Balancer (SLB) instances, and elastic IP addresses (EIPs).

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Asset Management > Tenant Assets**.
3. Select the required service. Example: **Elastic Compute Service (ECS)**.
4. Specify search conditions to view a specific asset.

   > **Note** If you want to view all assets, skip this step.

Search condition	Description
Department	The department to which the asset belongs.
VPC	The virtual private cloud (VPC) to which the asset belongs.
Status	The running status of the asset.
New	Specifies whether the asset to query is newly added.
Server name or IP address	The keywords of the asset name.

5. View asset information in the asset list.

27.1.9.3.2. View platform assets

This topic describes how to view the assets of the platform.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Asset Management > Platform Assets**.
3. Specify search conditions to view a specific asset.

   > **Note** If you want to view all assets, skip this step.

4. View asset information in the asset list.

27.1.9.4. Log Analysis

27.1.9.4.1. View the Log Overview page

This topic describes how to view the logs that are displayed in the widgets on the Log Overview page.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Log Overview**.
3. On the **Log Overview** page, view the widgets of the logs.

   You can perform the following operations to **modify** or **delete** the widgets on the **Log Overview** page:

   - To modify a widget, click **Modify** in the upper-right corner of the widget.
     a. In the **Modify** dialog box, reconfigure the **Chart Type**, **Category**, and **Value** parameters. When you configure the **Category** and **Value** parameters, take note of the following points:
        - **Category**: If you set **Chart Type** to **Bar Chart**, **Line Chart**, **Pie Chart**, or **Sheet**, you must specify this parameter.
        - **Value**: If you set **Chart Type** to **Pie Chart** or **Individual Value Plot**, you must specify this parameter.

     b. Click **Refresh** to preview the widget in the right side of the Modify dialog box.
     c. Above the widget, enter a new name to rename the widget.
     d. Click **OK**. The widget is updated on the **Log Overview** page.
   - To delete a widget, click **Delete** in the upper-right corner of the widget.

   The widgets on the **Log Overview** page are created on the **Log Audit** page. To create a widget, click **Please go to the log audit page to add a chart** in the **Add custom visualization chart** section in the lower part of the **Log Overview** page. On the **Log Audit** page, create a custom widget. For more information, see View global logs.

27.1.9.4.2. View global logs

This topic describes how to view global logs. Global logs are classified into logs of tenants, logs of platforms, and logs of data centers based the department to which the logs belong.

View a log widget

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Log Audit**.
3. Click the tab on which you want to view the logs of a department. For example, if you want to view the logs of tenants, click the **Cloud Tenant Logs** tab.
4. Specify search conditions, click the icon, and then view the logs that meet the search conditions in the sections of log distribution chart and log list.

Search condition	Description
Department	If you want to view the logs of a tenant, you can specify this search condition. Select the department to which the tenant belongs.
Log Source	Select the type of the system of which you want to collect the logs, the name of the system of which you want to collect the logs, and the log type from the drop-down list.
Duration	Select the time range within which you want to view the logs. Valid values: **Last 15 Minutes**, **Last 30 Minutes**, **Last 1 Day**, **Last 7 Days**, **Last 30 Days**, and **Custom**.
Start Time	Specify the start and end time within which you want to view the logs.
Log Content	Enter the log content in the search box. To save the search conditions as frequently used search conditions, click **Save Search Condition**. If you want to use these frequently used search conditions, click **Historical Records** to view the logs that meet the search conditions.

If you want to search for logs by using JSON domain-specific language (DSL) statements, click **Advanced Search**. In the **Advanced Search** dialog box, enter JSON DSL statements and click **Submit**.

Create a log widget

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Log Audit**.
3. Click the tab on which you want to view the logs of a department. For example, if you want to view the logs of tenants, click the **Cloud Tenant Logs** tab.
4. In the lower part of the Log Audit page, click the **Visualization** tab.
5. On the **Visualization** tab, configure parameters in the **Chart** section.

Parameter	Description
Chart Type	The type of widget to be displayed on the **Overview** page. Valid values: **Bar Chart**, **Line Chart**, **Pie Chart**, **Individual Value Plot**, and **Sheet**.
Category	This parameter is required when you select **Bar Chart**, **Line Chart**, **Pie Chart**, or **Sheet** for **Chart Type**. The type of items that you want to display in the horizontal axis or the column header of the widget.
Value Category	The type of items that you want to display in the vertical axis or the row header of the widget.

Parameter	Description
Value Type	The value type that you want to display in the widget. Valid values: **count**, **max**, **min**, **avg**, **sum**, **unique_count**, and **median**. If you want to display multiple value types in the widget, click **Add Value Field**. Then, you can configure the **Value Category** and **Value Type** parameters.

6. In the upper-left corner of the Visualization tab, enter a name for the widget.
7. Click **Refresh**.

   After the widget is created, you can view the widget on the **Overview** page.

   To configure the content to be displayed in the log list, you can perform the following steps: In the lower part of the Log Audit page, click the **Log** tab. Then, click the icon. To export the log list, click the icon.

27.1.9.4.3. Log configurations

27.1.9.4.3.1. Manage log sources

This topic describes how to view and manage the log sources that are connected to Security Operations Center (SOC).

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Log Configurations**.
3. Click the **Log Sources** tab. On the **Log Sources** tab, view the log overview information and log list.

   This tab provides the following information:

   - In the upper section of the **Log Sources** tab, you can view the total volume of log data and the total number of connected log sources.
   - In the upper-right corner above the log source list, you can specify the following search conditions to search for specific log sources.

Search condition	Description
Access System Types	The type of the log source that is connected to SOC. Valid values: **Host**, **Storage**, **Application**, **Networking**, **Data**, **Security**, and **Other**.
Log Types	The type of log that is collected by SOC. Valid values: **Operations Log**, **Operational Log**, **Alert Log**, and **Others**.
Access Mode	The mode that is used to collect logs. Valid values: **Custom** and **Built-in**.
Statuses	The status of the log source that is connected to SOC. Valid values: **On** and **Off**. In the log source list, find a log source and in the **Status** column click the icon to set the log source to on or off. In the Tips message, click **OK**.

4. Find a log source and click **Modify** in the Actions column.
5. In the **Edit** dialog box, configure the Storage Days and View Permission parameters and click **OK**.

27.1.9.4.3.2. Create a log collection task

This topic describes how to create a log collection task.

Prerequisites

If you use Logtail to collect logs, ensure that the following conditions are met:

- Logtail is installed.
- A server group that you want to configure Logtail is created.

For more information, see Manage log collectors.

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Log Configurations**.
3. On the **Log Configuration** page, click the **Log Access Task** tab.
4. On the **Log Access Task** tab, click **Add**.
5. In the **Configure Log Source** step of the **Create Task** wizard, configure parameters and click **Next**.

Parameter	Description
Task Name	The name of the log collection task.
Access System Name	The name of the log source from which you want to collect the logs. Enter the name of the log source, such as Windows operating system.
Access System Type	The type of log source from which you want to collect the logs. Valid values:**Host**, **Storage**, **Application**, **Networking**, **Data**, **Security**, and **Other**.
Log Source Name	The name of the log source.
Log Type	The type of the logs that you want to collect. Valid values: **Operations Log**, **Operational Log**, **Alert Log**, and **Others**.
Source	The department to which the logs belong. Valid values: **Cloud Tenant**, **Cloud Platform**, and **On-Premises Data Center**.

6. In the left-side navigation tree of the **Configure Access Mode** step, select a data source type, configure parameters, and then click **Next**.

   Valid values of data source types: Syslog, SLS, and Logtail.

   - If you select Syslog, you must configure the following parameters.

Parameter	Description
IP address	The IP address or Classless Inter-Domain Routing (CIDR) block used to report syslog logs.
Protocol	The network protocol used when the logs of the log source are collected. Valid values **UDP** and **TCP**.
Access System Type	The type of log source from which you want to collect the logs. Valid values: **Host**, **Application**, **Networking**, and **Other**.
Keyword	The keyword of the log source.

   - If you select SLS, you must configure the following parameters.

Parameter	Description
Log Project	The name of the Log Service project.
Log Store	The name of a Logstore in Log Service.
Endpoint	The endpoint used to connect to the Log Service project.
accessKey	The AccessKey ID of your account to access the Log Service project.
secretKey	The AccessKey secret of your account to access the Log Service project.

- If you select Logtail, you must configure the following parameters.

Parameter	Description
Name	The name of Logtail.
Log Type	The type of the logs that you want to collect. Valid values: **JSON**, **Apsara**, **Separator**, and **Regular Expression**.
Log Pattern	The format of log file names. Example: *access*.log*.
Log Path	The path of the logs that you want to collect. Absolute paths and relative paths are supported. You can use wildcards in relative paths.
Log Sample	A sample log entry from the logs that you want to collect.
Regular Expression to Match First Log Entry	After you enter the **sample log entry**, click **Generate Automatically**. A regular expression is generated to match the first line of the log entry.
Regular Expression	Click **Regular Expression**. In the **Generate Regular Expression** dialog box, select the fields that you want to extract and click **Generate Regular Expression**. After the regular expression is generated, click **OK**.
Date Format	The date format that is automatically generated based on the extracted time fields.
Apply to Server Group	The server group for which you want to configure Logtail.

7. In the **Parse and Normalize Data** step, configure parameters and click **Next**.

Parameter	Description
Data Acquisition Method	Select the acquisition method used to query the sample log entry. Valid values: **Automatic Acquisition** and **Manual Input**.
Sample Data	If you set **Data Acquisition Method** to **Manual Input**, you must enter the sample log entry.
Parser	Select a parser based on the sample log entry. Valid values: **JSON** and **jsonArray**.

Parameter	Description
Extract Value	The content that can be extracted from the logs. Click **Add Field**. In the **Add Field** dialog box, configure the **Original Extract Field**, **Original Extract Value**, **Enrich Data**, and **Target Type** parameters, and click **OK**. Valid values of **Enrich Data**: **Retain Original Field**, **Field Masking**, **Delete Field**, **Rename Field**, **Automatically Fill System Time**, **Assign Value to Constant**, and **Assign Value to Variable**.

8. In the **Data Output** step, configure the **Log Source Name**, **Storage Duration**, and **View Permissions** parameters.
9. Click **Save**.

   After the log collection task is created, the task appears in the list for log collection tasks. You can publish, modify, and delete the task in the Actions column of the task.

   - **Publish**: After the log collection task is created, logs are not automatically collected. You must click **Publish**. After you publish the task, the  icon appears in the **Access Status** column of the task.

     You can click the switch in the **Access Status** column to change the status of the log collection task.
   - **Edit**: You can click **Edit** to change the configurations in the **Parse and Normalize Data** and **Data Output** steps. You cannot change the data source type and parser.
   - **Delete**: If you no longer need the log collection task, click **Delete** in the Actions column of the task.

27.1.9.4.3.3. Manage log collectors

Before you can use Logtail to collect logs, you must install Logtail, add log collectors, and create server groups. This topic describes how to install Logtail, add log collectors, and create server groups.

Install Logtail

If Logtail is installed, the system automatically uninstalls the existing version of Logtail, deletes the */usr/local/ilogtail* directory, and then reinstalls Logtail. After the new version of Logtail is installed, Logtail automatically runs when the system starts.

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Log Configurations**.
3. On the **Log Configurations** page, click the **Collectors** tab.
4. Install Logtail.

   On the **Collectors** tab, install Logtail based on the operating system of the server and the installation method that is supported by Logtail.

   - If the server runs Linux and Logtail supports manual installation, perform the following steps:
     a. In the **Linux Operating Systems** section, click **Download Installation Package** and specify whether to download a 32-bit or 64-bit installation file to your local computer.
     b. Log on to the server as an administrator. Then, run the installation command to install Logtail.
   - If the server runs Windows and Logtail supports manual installation, perform the following steps:
     a. In the **Windows Operating Systems** section, click **Download Installation Package** to download the installation file to your local computer.

b. Upload the installation file to the server. For example, you can use an FTP client to upload the installation file to the server.

c. Run the installation file on the server as an administrator.

After Logtail is installed, Logtail automatically runs when the system starts.

Add a log collector

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Log Configurations**.
3. In the **Collectors** section, click **Add Collector**.
4. In the **Add Collector** dialog box, configure the following parameters and click **OK**.

Parameter	Description
Collector IP Address	The IP address of the log source whose logs you want to collect.
Server Name	The name of the server where Logtail is installed.
Operating System	The operating system of the server where Logtail is installed.

After the log collector is added, you can view the information of the log collector in the **Collectors** section. The information includes **Collector IP Address** and **Server Name**.

If you want to view the details of Logtail, find the log collector and click **View Logtail Configurations** in the Actions column. In the **Configure Logtail** panel, you can view the details of Logtail.

Create a server group

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Log Configurations**.
3. In the **Server Groups** section, click **Create Server Group**.
4. In the **Create Server Group** dialog box, configure the following parameters and click **OK**.

Parameter	Description
Server Group Name	The name of the server group.
IP Addresses	The IP address of the server that you want to add to the server group. To add a server to the server group, perform the following steps: i. In the **Ungrouped Servers** section, select the server that you want to add to the group. ii. Click the > icon to add the server to the **Servers to Be Grouped** section.

After you create the server group, you can view the information of the server group in the **Server Groups** section. If you want to view the details of a server group or delete a server group, perform the following steps:

- Find the server group. Then, click **Details** in the Actions column to view the details of the server group.
- Find the server group. Then, click **Delete** in the Actions column to delete the server group.

27.1.9.4.3.4. Manage storage policies

This topic describes how to view and configure storage policies for logs.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Log Configurations**.
3. On the **Log Configurations** page, click **Policy Management**.
4. On the **Storage Policy** tab of the **Policy Management** page, view the storage policies for logs.
5. Click **Edit**.
6. On the **Storage Policy** tab, configure parameters in the Log Storage (Advanced) and Security Monitoring Data Storage (Advanced) sections and click **Modify**.

Section	Parameter	Description
Log Storage (Advanced)	Maximum Log Size	The upper limit of the storage space that can be occupied by logs.
	Maximum Storage Period	The maximum number of days during which logs can be stored.
Security Monitoring Data Storage (Advanced)	ApsaraDB RDS Storage Period	The maximum number of days during which security monitoring data can be stored.

27.1.9.4.4. Security Audit

27.1.9.4.4.1. Overview

A security audit refers to the systemic and independent inspection and verification of activities and behavior in the computer network environment. Delegated by property owners and authorized by management authorities, professional auditors give their assessments according to relevant laws and regulations. When the administrator needs to backtrack system operations, the administrator can perform a security audit.

Security audits are long-term security management activities throughout the lifecycle of cloud services. The security audit feature of Apsara Stack Security can collect system security data, analyze weaknesses in system operations, report audit events, and classify audit events into important, moderate, and low risk levels. The security administrator views and analyzes audit events to continuously improve the system and ensure the security and reliability of cloud services.

27.1.9.4.4.2. View security audit overview

This topic describes how to view the summarized information about security audit.

Context

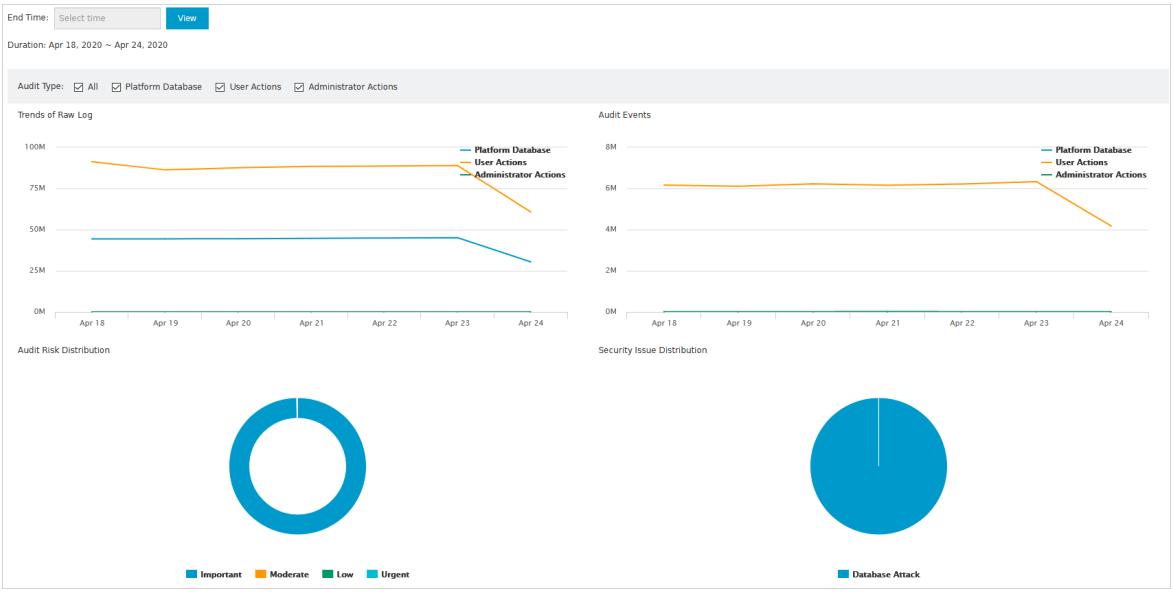
The **Overview** tab provides reports on the raw log trend, audit event trend, audit risk distribution, and security event distribution. The reports are displayed in run charts or pie charts to help security administrators analyze the trend of risks in your cloud services.

On the **Overview** tab, security administrators can check the number of log entries and the storage usage in a specific time range.

Procedure

1.

2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the page that appears, click the **Overview** tab.
3. On the **Overview** tab, view the audit summary for the last seven days.



   - Trends of Raw Log

     This chart displays the trend of logs generated by physical servers, network devices, ApsaraDB RDS instances, Elastic Compute Service (ECS) instances, and API calls in the last seven days. Security administrators can analyze the trend to check whether the number of log entries is at a normal level.

   - Audit Events

     This chart displays the trend of audit events that are generated by physical servers, network devices, ApsaraDB RDS instances, ECS instances, and API calls in the last seven days. Security administrators can analyze the trend to check whether the number of audit events is at a normal level.

   - Audit Risk Distribution

     This chart displays the percentage distribution of audit events at different risk levels in the last seven days. Risk levels are important, moderate, and low. Security administrators can analyze the trend to check whether the audit events are at acceptable risk levels.

   - Security Issue Distribution

     This chart displays the percentage distribution of different event types in the last seven days. Security administrators can analyze this chart to check for the most frequent audit events and identify high-risk events to improve security protection.

   - Log Size

     This chart displays the volume of online logs and offline logs. If these logs consume many storage resources, we recommend that you back up required audit logs and delete unnecessary logs.

   - Audit Log Size

     This chart displays the size of logs for each audit type.

4. View the audit summary in a specific time range.
   i. Specify **End Time** as the end of the time range to query.
   ii. In **Audit Type**, select the audit types to query.
   iii. Click **View** to view the audit summary in the last seven days before the specified end time.

27.1.9.4.4.3. Query audit events

This topic describes how to query audit events.

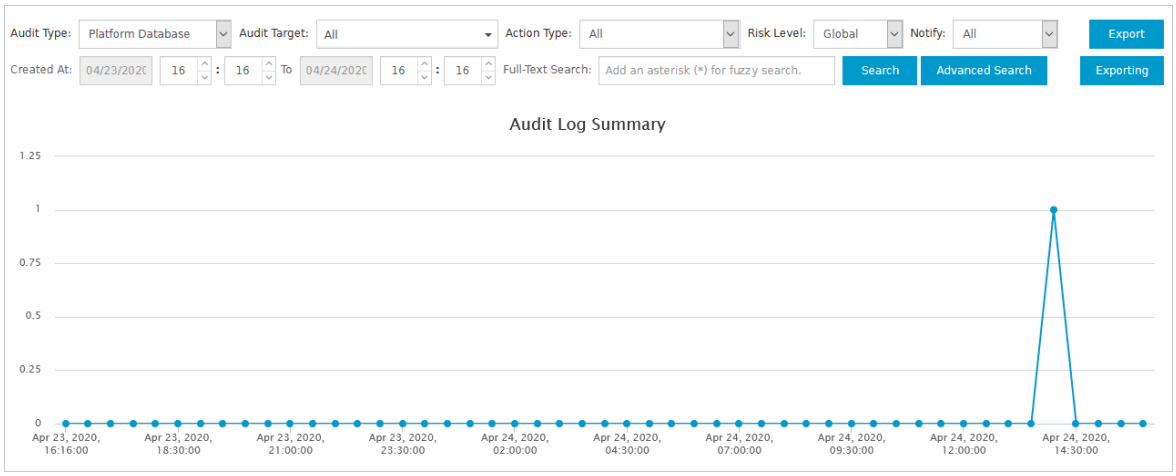
Context

On the **Audit Query** tab, you can view the details of audit events, including the log creation time, audit type, audit object, action type, risk level, and log content.

The system matches the logs that are collected by a security audit module with audit rules. If the log content matches the regular expression in an audit rule, an audit event is reported. For more information about audit rules, see Add an audit policy.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the page that appears, click the **Audit Query** tab.
3. On the **Audit Query** tab, configure query conditions to view audit events within the specified time range.



   - Basic query
     a. Configure **Audit Type**, **Audit Target**, **Action Type**, **Risk Level**, and **Notify**.
     b. Specify a time range to query.
     c. In the **Full-Text Search** search box, enter a keyword.
     d. Click **Search**.
   - Advanced query

     In addition to the basic query conditions, you can configure advanced query conditions.
     a. Configure basic query conditions.
     b. Click **Advanced Search**.
     c. Below **Filter Condition**, configure **User**, **Target**, **Action**, **Result**, and **Cause**.
     d. Click **Save**.
4. Click **Export** to export the audit events.

   Download the exported file for analysis. For more information, see Manage export tasks.

27.1.9.4.4.4. View raw logs

This topic describes how to view raw audit logs.

Context

On the **Raw Log** tab, you can view the raw logs generated by a running audit object. Raw logs contain

information that is required for debugging. Security administrators can use these raw logs to troubleshoot system failures.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the page that appears, click the **Raw Log** tab.
3. On the **Raw Log** tab, configure query conditions to view the log summary chart and raw logs within a specific time range.
   i. Specify **Audit Type** and **Audit Target**.
   ii. Enter a keyword.
   iii. Specify a time range to query.
   iv. Click **Search**.
4. Click **Export** to export the data.

   Download the exported file for analysis. For more information, see Manage export tasks.

27.1.9.4.4.5. Manage log sources

This topic describes how to view and manage log sources.

Context

You can view the number of log entries by log type or log source. You can also specify whether to display logs.

- The Log Types sub-tab provides the number of all log entries for a specific audit object of a specific device instance.
- The Log Sources sub-tab provides the number of log entries for all audit objects of a specific device instance.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the page that appears, click the **Log Sources** tab.
3. Click the **Log Types** tab and view the number of log entries for each audit object.

   You can view the number of log entries that are recorded on the current day and the number of log entries that are recorded during the last 30 days for each audit object.

   If you do not want to display the log entries for an audit object, perform the following steps:

   i. Find the audit object and click **Hide** in the **Actions** column.
   ii. In the Note message, click **Confirm**.

   > **Note** The process that is used to display the log entries for an audit object is similar to the process that is used to hide the log entries.

4. Click the **Log Sources** sub-tab and view the number of log entries for each device instance.

   You can view the number of log entries that are recorded on the current day and the number of log entries that are recorded during the last 30 days for each device instance.

   If you do not want to display the log entries for an audit object from a specific device instance, perform the following steps:

   i. Find the device instance and click **Hide** in the **Actions** column.
   ii. In the Note message, click **Confirm**.

> **Note** The process that is used to display the log entries for an audit object is similar to the process that is used to hide the log entries.

27.1.9.4.4.6. Policy settings

Manage audit rules

This topic describes how to create, modify, or delete an audit rule.

Context

If a log entry matches an audit rule, an audit event is reported. You can specify regular expressions in an audit rule to match log entries. A regular expression defines a matching pattern for character strings and can be used to check whether a string contains a specific substring. The following table provides examples about the pattern.

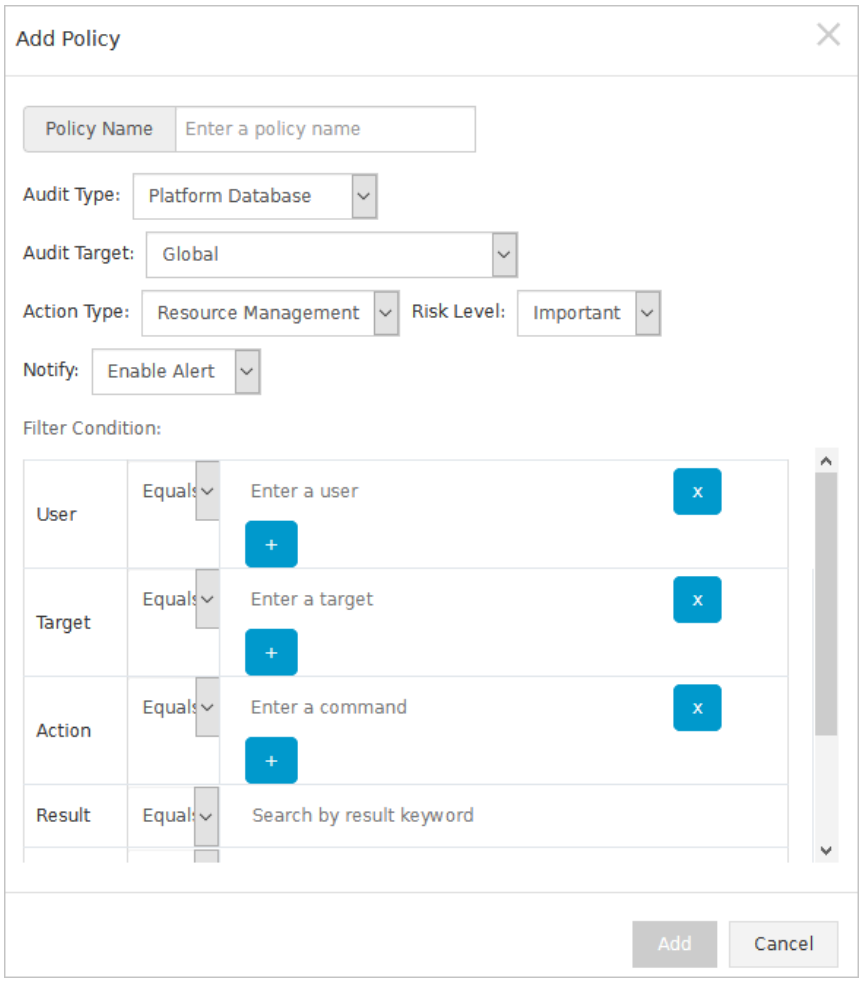
Regular expression	Description
`^\d{5,12}$`	Matches the consecutive numbers from the fifth number to the twelfth number.
`load_file\(`	Matches the "load_file(" string.

The security audit module defines the default audit rule based on the string that is generated in the log. This applies when an audit event is reported. The security administrator can also customize audit rules based on the string that is generated in the log. This applies when the system encounters an attack.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the page that appears, click the **Policies** tab.
3. On the **Policies** tab, click the **Audit Rules** sub-tab.
4. Create an audit rule.
   i. Click **New** in the upper-right corner.

ii. In the **Add Policy** dialog box, configure parameters.



iii. Click **Add**.

The system sends an alert email to the specified alert recipient after you create an audit rule. This applies if one string in an audit log of the specified audit type, audit object, or risk level matches the regular expression of the audit rule.

5. Manage audit rules.

   You can create, query, disable, enable, and delete audit rules.

   - Query audit rules

     Specify **Audit Type** and **Audit Target**. Enter a keyword in the search box and click **Search**.

   - Disable an audit rule

     Find the audit rule that you want to disable and click **Disable** in the **Actions** column.

   - Enable an audit rule

     Find the audit rule that has been disabled and click **Enable** in the **Actions** column.

   - Delete an audit rule

     Find the audit rule that you want to delete and click **Delete** in the **Actions** column.

     > **Note** You can delete only custom rules.

Configure alert recipients

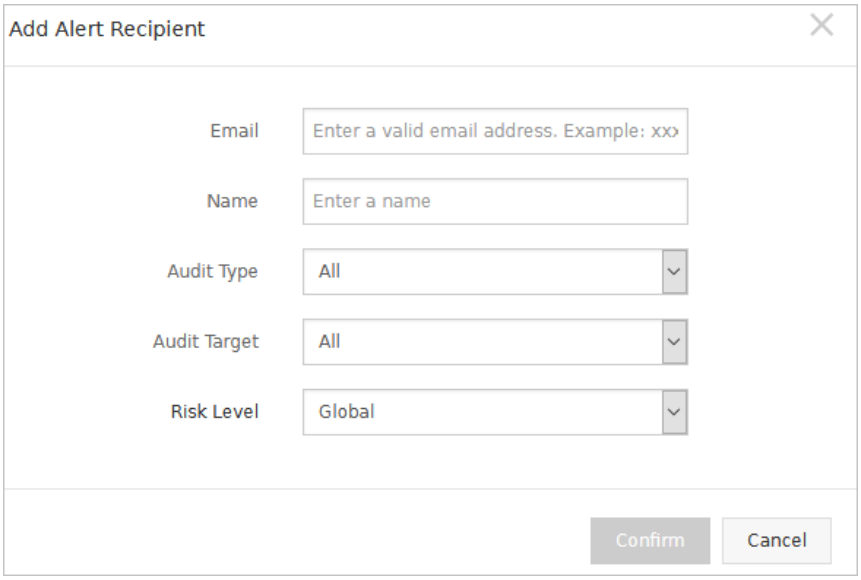
This topic describes how to configure the recipients of alerts on audit events.

Context

You can add an alert recipient by entering an email address that can be used to receive alerts on audit events.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the page that appears, click the **Policies** tab.
3. On the **Policies** tab, click the **Alert Settings** sub-tab.
4. Create an alert recipient.
   i. Click **New**.
   ii. In the **Add Alert Recipient** dialog box, configure parameters.



   iii. Click **Confirm**.
5. Manage alert recipients.
   - Search for alert recipients

     Specify **Audit Type**, **Audit Target**, and **Risk Level**, enter the keyword of the email address, and then click **Search**.
   - Delete alert recipients

     Find the email address that you want to delete and click **Delete** in the Actions column.

Manage archives of events and logs

This topic describes how to query and download the archives of audit events and raw logs.

Context

You can download the archives of events and logs to analyze audit events. This ensures the security of the Apsara Stack environment.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security**

Audit. On the page that appears, click the **Policies** tab.

3. On the **Policies** tab, click the **Archiving** sub-tab.
4. Query the archives of events and logs.
   i. Specify **Audit Type** and **Archiving Type**.
   ii. Specify a time range to query.
   iii. Click **Search**.
5. Find the file where the archive information is stored and click **Download** in the **Actions** column to save the archive file to your on-premises machine.

Manage export tasks

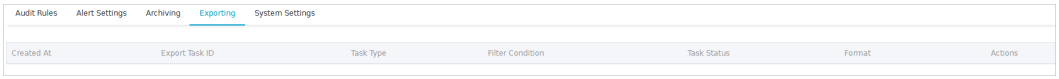
This topic describes how to download or delete exported audit events and logs.

Context

You can export audit events or logs on the **Audit Query** or **Raw Log** tab of the Security Audit page. After you export audit events or logs, you can manage the export tasks on the Exporting sub-tab.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the page that appears, click the **Policies** tab.
3. On the **Policies** tab, click the **Exporting** sub-tab.
4. View the created export tasks.



5. Click **Download** to download audit events or logs to your on-premises server.
6. Click **Delete** to delete the export task.

Modify system settings

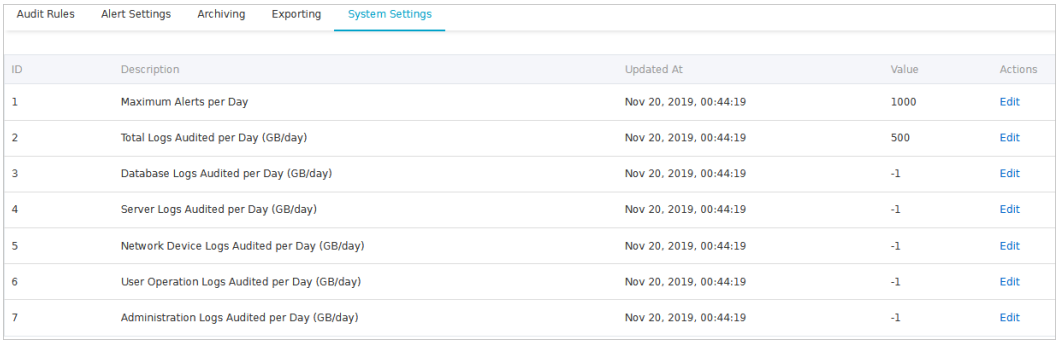
This topic describes how to configure system parameters for security audit.

Context

You can configure system parameters to specify the maximum number of system alerts per day and the maximum number of audits per day for raw logs.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the page that appears, click the **Policies** tab.
3. On the **Policies** tab, click the **System Settings** sub-tab.
4. Find the configuration item that you want to modify and click **Edit** in the **Actions** column.



5. Enter a required value in the Value column and click **Confirm** in the Actions column.

27.1.9.5. Rules

27.1.9.5.1. Create an IPS rule for traffic monitoring

This topic describes how to create an intrusion prevention system (IPS) rule for traffic monitoring in Cloud Firewall. Cloud Firewall has built-in IPS rules. This topic describes how to customize IPS rules based on your business requirements and network environment.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Rules**.
3. On the Rules page, click the **Cloud Firewall IPS Rules** tab.
4. Click **Create Rule**.
5. In the **Create Rule** panel, configure parameters.

Parameter	Description
Rule Name	The name of the IPS rule. We recommend that you enter an informative name for easy management.
Rules Engine	The rules engine. Valid values: **Basic Policies** and **Virtual Patches**.
Attack Type	The type of attack to be detected by the rule.
Severity	The severity. Valid values: **Low**, **Medium**, and **High**.
CVE	The Common Vulnerabilities and Exposures (CVE) ID of the vulnerability listed in the rule. **Note** CVE provides a list of the public security vulnerabilities. CVE IDs are allocated by a CVE Numbering Authority (CNA).
Application	The name of the attacked application.
Rule Mode	The rule mode. Valid values: **Package** and **Traffic**.
Direction	The direction of traffic to be monitored by the IPS rule. Valid values: **Inbound and Outbound**, **Inbound**, and **Outbound**.

Parameter	Description
Rule Content	The rule content that is specified by using the Snort syntax. **Note** To avoid negative impacts on your business, make sure that you enter valid content for the rule.
Rule Description	The rule description. We recommend that you enter information such as the purpose and impact of the rule.
Description	The remarks for the rule. We recommend that you enter information such as the purpose and impact of the rule.

6. Click **OK**.

27.1.9.5.2. Manage IPS rules of Cloud Firewall

This topic describes how to view, enable, and disable the intrusion prevention system (IPS) rules of Cloud Firewall.

Context

On the **Cloud Firewall IPS Rules** tab of the Rules page in Apsara Stack Security Center, you can view the built-in and custom IPS rules, and enable or disable the rules based on your business requirements.

Procedure

1.
2. Choose **Global Platform Security > Alibaba Cloud Security**.
3. On the Rules page, click the **Cloud Firewall IPS Rules** tab.
4. Manage IPS rules of Cloud Firewall.

   In the list of IPS rules, you can view rule details, enable rules, and disable rules.

   - View rule details

     Find the rule whose details you want to view and click **Details** in the **Actions** column to view the rule details.

   - Enable a rule

     Find the rule that you want to enable and turn on the switch in the **Enable or not** column to change the status of the rule from **Disable** to **Enable**.

   - Disable a rule

     If a rule is not suitable for your business, you can disable the rule.

     Find the rule that you want to disable and turn off the switch in the **Enable or not** column to change the status of the rule from **Enable** to **Disable**.

27.1.9.5.3. Create IDS rules for traffic monitoring

This topic describes how to create intrusion detection system (IDS) rules for traffic monitoring.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Rules**.
3. On the Rules page, click the **Traffic Monitoring IDS Rules** tab.

4. Click **Create Rule**.
5. In the **Create Rule** panel, configure the following parameters.

Parameter	Description
Rule Name	The name of the IPS rule. We recommend that you enter a name that can help you identify and manage the IPS rule in an efficient manner.
Rules Engine	The rules engine that you want to use. Valid values: **Basic Policies** and **Virtual Patches**.
Attack Type	The type of the attack that you want to detect by using the IPS rule
Severity	The severity of the attack. Valid values: **Low**, **Medium**, and **High**.
CVE	The Common Vulnerabilities and Exposures (CVE) ID of the vulnerability that you want to add to the rule. **Note** CVE provides a list of public security vulnerabilities. CVE IDs are allocated by a CVE Numbering Authority (CNA).
Application	The name of the attacked application.
Rule Mode	The mode of the IPS rule. Valid values: **Packet** and **Traffic**.
Direction	The direction of traffic that you want to monitor by using the IPS rule. Valid values: **Inbound and Outbound**, **Inbound**, and **Outbound**.
Rule Content	The content of the IPS rule. You must use the Snort syntax to specify the content. **Note** To prevent a negative impact on your business, make sure that the content you enter for the IPS rule is valid.
Rule Description	The description of the IPS rule. We recommend that you enter information that can help you identify the IPS rule, such as the purpose or impact of the rule.
Description	The additional description of the IPS rule. We recommend that you enter information that can help you identify the IPS rule, such as the purpose or impact of the rule.

6. Click **OK**.

27.1.9.5.4. Manage IDS rules for traffic monitoring

This topic describes how to view, enable, and disable intrusion detection system (IDS) rules for traffic monitoring.

Context

On the **Traffic Monitoring IDS Rules** tab, you can view the built-in and custom IDS rules. It can also be used to enable or disable the rules based on your business requirements.

Procedure

1.
2. Choose **Global Platform Security > Alibaba Cloud Security**.

3. On the Rules page, click the **Traffic Monitoring IDS Rules** tab.
4. Manage IDS rules for traffic monitoring.

   In the IDS rule list, you can view rule details, enable rules, and disable rules.

   - View rule details

     Find the rule whose details you want to view and click **Details** in the **Actions** column to view the rule details.

   - Enable a rule

     Find the rule that you want to enable and turn on the switch in the **Enable or not** column to change the status of the rule from **Disable** to **Enable**.

   - Disable a rule

     If a rule is not suitable for your business, you can disable the rule.

     Find the rule that you want to disable and turn off the switch in the **Enable or not** column to change the status of the rule from **Enable** to **Disable**.

27.1.9.5.5. Customize DDoS traffic scrubbing policies and traffic redirection thresholds

This topic describes how to customize DDoS traffic scrubbing policies and traffic redirection thresholds. Default traffic redirection thresholds are provided. If you want to customize the thresholds, perform the following steps.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Rules**.
3. i. Choose **AliGuard Rules > Scrubbing Policy**.
   ii. Find a specific rule and click **Modify Threshold** in the **Actions** column
   iii. In the **Modify Threshold** dialog box, enter a threshold value.
   iv. Click **OK**.
4. Customize the traffic redirection threshold.
   i. Choose **AliGuard Rules > Reroute Threshold**.
   ii. Find a specific rule and click **Modify Threshold** in the **Actions** column.
   iii. In the **Modify Threshold** dialog box, enter a threshold value.
   iv. Click **OK**.

27.1.9.5.6. View Server Guard rules

This topic describes how to view the operations of Server Guard rules. You can view the list of vulnerabilities, baselines, and host exceptions.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Rules**. On the page that appears, click the **Server Guard Rules** tab.
3. In the overview section, you can view the total number of **vulnerability libraries**, number of **baselines**, and number of **host exceptions** as well as the available **engines**.
4. View the vulnerability list.

i. Click the **Vulnerabilities** tab.

ii. In the overview section, you can view the total number of **Linux vulnerabilities**, total number of **Windows vulnerabilities**, total number of **Web-CMS vulnerabilities**, and total number of **urgent vulnerabilities**.

iii. Specify search conditions to view the vulnerabilities that meet the search conditions.

> **Note** If you want to view all vulnerabilities, skip this step.

In the vulnerability list, you can view the **vulnerability name**, **CVE ID**, **vulnerability type**, **system**, **update time**, and **status**.

5. View the baseline list.

i. Click the **Baselines** tab.

ii. In the overview section, you can view the numbers of baseline types and check items.

iii. Specify search conditions to view the baselines that meet the search conditions

> **Note** If you want to view all baselines, skip this step.

In the baseline list, you can view the **baseline type**, **check item category**, **check item name**, **risk level**, **update time**, and **status**.

6. View the host exception list.

i. Click the **Server Exceptions** tab.

ii. In the overview section, you can view the number of **rule alert subcategories**, number of webshells, and number of malicious viruses.

iii. Specify search conditions to view the host exceptions that meet the search conditions.

> **Note** If you want to view all exceptions, skip this step.

In the host exception list, you can view the **subcategory name**, **rule category**, **risk level**, **update time**, **source**, and **status**.

27.1.9.6. Threat intelligence

27.1.9.6.1. Enable the service configuration feature

The threat intelligence module integrates threat monitoring and big data analysis. This can be used to obtain the latest information about developments in the threat intelligence field. After you enable the service configuration feature, the system starts to monitor and collect threat intelligence. This topic describes how to enable the service configuration feature.

Procedure

1.

2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Threat Intelligence > Service Configuration** .

3. On the **Service Configuration** page, view the data types and descriptions on the Situation Awareness and Web application firewall tabs.

4. Click the tab where you want to enable threat monitoring and turn on **Activation status**.

   After you turn on **Activation status**, the system starts to monitor and collect threat intelligence for the data types listed on the tab.

What's next

After you enable the service configuration feature, choose **Security Operations Center (SOC) > Threat Intelligence**. On the **Overview** page, you can view the overall situation and statistics of threats during the last 30 days. For more information, see View the Overview page.

27.1.9.6.2. View the Overview page

The Overview page displays the overall situation and statistics of threats to your assets over the last 30 days.

Prerequisites

The **service configuration** feature is enabled. For more information, see Enable the service configuration feature.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Threat Intelligence > Overview**.
3. On the **Overview** page, view the statistics and threats that are detected on Apsara Stack services by the threat intelligence module.

   On the **Overview** page, you can perform the following operations:

   - View **Total malicious metric intelligence**

     In the **Total malicious metric intelligence** section of the **Overview** page, view the information about the detected threats on Apsara Stack services. The information includes the number of malicious IP addresses, malicious domain names, and malicious URLs.
   - View **Threat trends in the last 30 days**
   - Search for an IP address to check whether the IP address is malicious.

     Enter the IP address that you want to check in the search box in the upper-right corner of the IP Report page and click the icon. To view the details of the IP address, choose **Security Operations Center (SOC) > Threat Intelligence > IP Address Search**. On the page that appears, click the **IP Report** tab. For more information, see Search for an IP address.
   - View **Top 10 active IP malicious addresses**

     In the **Top 10 active IP malicious addresses** section of the **Overview** page, view the information about the top 10 malicious IP addresses. The information includes **IP address**, **First malicious observation**, **Last malicious observation**, and **Malicious label**.

27.1.9.6.3. Search for and view the information about a suspicious or malicious IP address

The threat intelligence module allows you to search for threat intelligence. This module helps you handle suspicious or malicious IP addresses at the earliest opportunity.

Prerequisites

The **service configuration** feature is enabled. For more information, see Enable the service configuration feature.

Procedure

1.

2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Threat Intelligence > IP Address Search**.
3. In the search box on the **Search** page, enter the suspicious or malicious IP address that you want to query and click the icon.
4. On the **IP Report** page, view **Threat Level**, **Basic Information**, **Threat Overview**, **IP Details**, and **Attack Risk Level Analysis** of the IP address.

   You can view the following information on the IP Report page:

   - **Threat Level**: View the threat level of the suspicious or malicious IP address.

     The threat intelligence module classifies IP addresses into three threat levels. The levels are normal, suspicious, and high-risk. If the IP address is identified as high-risk, we recommend that you handle the IP address at the earliest opportunity.

   - **Basic Information**: View the basic information about the suspicious or malicious IP address.

     The basic information includes the server in a data center, Abstract Syntax Notation One (ASN.1), country and city to which the IP address belongs, and the number of domain names for the IP address.

   - View the statistics of the suspicious or malicious IP address.

     You can view **Threat Overview**, **IP Details**, and **Threat Details** of the IP address.

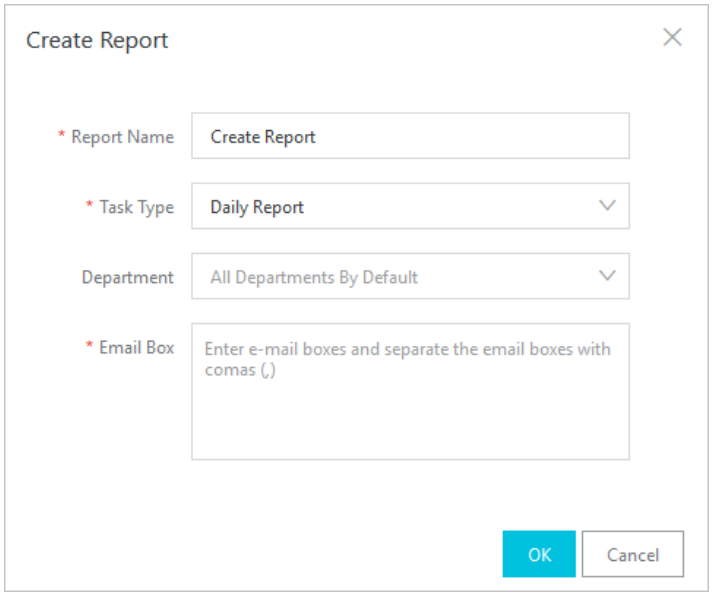
     - The **Threat Overview** tab displays **Top 5 Attack Preference**, **Attack Number**, and **Attack Level Analysis** of the IP address.
     - The **IP Details** tab displays **WHOIS** and **IP reverse check information** of the IP address.
     - The **Threat Details** tab displays the **threat tag** list of the IP address. The list contains specific information. The information includes the intelligence source, time when the IP address is detected for the first time, time when the IP address is active for the last time, and threat tag.

27.1.9.7. Create a report task

This topic describes how to create a report task. After you create a report task, the system sends reports on a regular basis.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Report Management**.
3. On the Report Management page, click **Create Report**.
4. In the **Create Report** dialog box, configure parameters.



Parameter	Description
Report Name	The name of the report task. We recommend that you enter information such as the report purpose for easier identification and management.
Task Type	The type of the task. Valid values: **Daily Report**, **Weekly Report**, and **Monthly Report**.
Department	The department related to the report.
Email Box	The email address of the report recipient. If you enter more than one email address, separate the email addresses with commas (,).

5. Click **Confirm**.

Result

In the report task list, you can view, edit, and delete the newly created report tasks.

27.1.9.8. System Configurations

27.1.9.8.1. View and manage metrics

Apsara Stack Security Center allows you to monitor security services. This helps find performance bottlenecks at the earliest opportunity. Then, you can scale out, scale up, or downgrade services to prevent system failures. This topic describes how to view the information about each security service in Apsara Stack Security Center and how to manage metrics.

View information about overall system monitoring

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Security Monitoring**.
3. On the **Overall System Monitoring** tab, view the overall information about each security service in Apsara Stack Security Center and the list of monitored security services.

   In the upper-left corner of the **Overall System Monitoring** page, you can view the overall information about security services.

- **Services**: the number of security services monitored by Apsara Stack Security Center.
- **Monitoring Items**: the number of metrics.
- **Abnormal Items**: the number of metrics whose status is abnormal.

In the list of monitored security services, you can view the following information.

Parameter	Description
Services	The security service monitored by Apsara Stack Security Center.
Metrics	The monitoring indicator for the monitored security service.
Description	The description of the monitoring indicator.
Monitoring Items	The number of metrics that belong to a monitoring indicator.
Abnormal Items	The number of metrics whose status is abnormal, and the numbers of metrics at each urgency level. The urgency levels are indicated by different colors. The red color indicates a critical exception, the orange color indicates an important exception, and the blue color indicates a moderate exception.
State	The running status of the monitoring indicator.

4. Click **Details** in the **Actions** column of a monitoring indicator. In the **Monitoring details** panel, view the details of the monitoring indicator.

   In the upper-left corner of the **Monitoring details** panel, you can view the overall information about the metrics.

   - **Total Monitoring Items**: the number of metrics that belong to the monitoring indicator.
   - **Normal Items**: the number of metrics in the normal state.
   - **Abnormal Items**: the number of metrics in the abnormal state.

   In the metric list, you can view the following information.

Parameter	Description
Monitoring Metrics	The name of the metric.
Adjust Alert Threshold	The threshold for the metric. If the value of the metric reaches the threshold and lasts for a specific period of time, the status of the metric becomes abnormal.
Duration	The time period. If the value of the metric reaches the threshold and lasts for the specified period of time, the status of the metric becomes abnormal.
Monitoring Level	The urgency level displayed when the status of the metric becomes abnormal.
Status	The status of the metric.
Alert Notifications	The switch in the **Alert Notifications** column. If the status of the metric becomes abnormal, you can turn on or off the switch.

Manage metrics

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Security Monitoring**.

3. On the **Overall System Monitoring** tab, click **Details** in the **Actions** column.
4. In the **Monitoring details** panel, click **Adjust Threshold**, **Handle**, or **Adjust Duration** in the **Actions** column to manage a metric.

   You can perform the following operations on the metric:

   - **Adjust Threshold**: In the **Adjust Threshold** dialog box, modify the threshold and click **OK**.

     After you modify the threshold, an alert is generated when the value of the metric reaches the new threshold.
   - **Handle**: You can configure the status of the metric based on your business requirements.
     - If you do not want to handle the alert generated from the metric, select **Ignore** from the drop-down list. The status of the metric is **Ignored**.
     - After you handle the alert generated from the metric, select **Handled** from the drop-down list. The status of the metric is **handled**.
   - **Adjust Duration**: In the **Adjust Duration** dialog box, modify the time and click **OK**.

     When the value of the metric reaches the threshold and lasts for the specified period of time, the status of the metric becomes abnormal.

27.1.9.8.2. Alert settings

27.1.9.8.2.1. Configure alert contacts

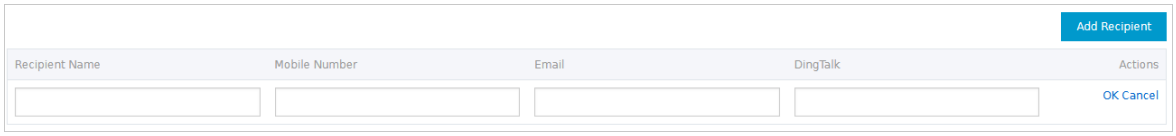
This topic describes how to configure and manage alert contacts.

Context

Apsara Stack Security sends alert notifications to alert contacts by text message, email, or DingTalk. When the detected information matches an alert rule, Apsara Stack Security sends an alert notification to the alert contacts.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Alert Settings**. On the page that appears, click the **Alert Recipient** tab.
3. Click **Add Recipient**.
4. Enter the contact information and click **OK**.



5. Manage alert contacts.

   In the contact list, find a contact and click **Edit** in the Actions column to edit the contact information.

27.1.9.8.2.2. Configure alert notifications

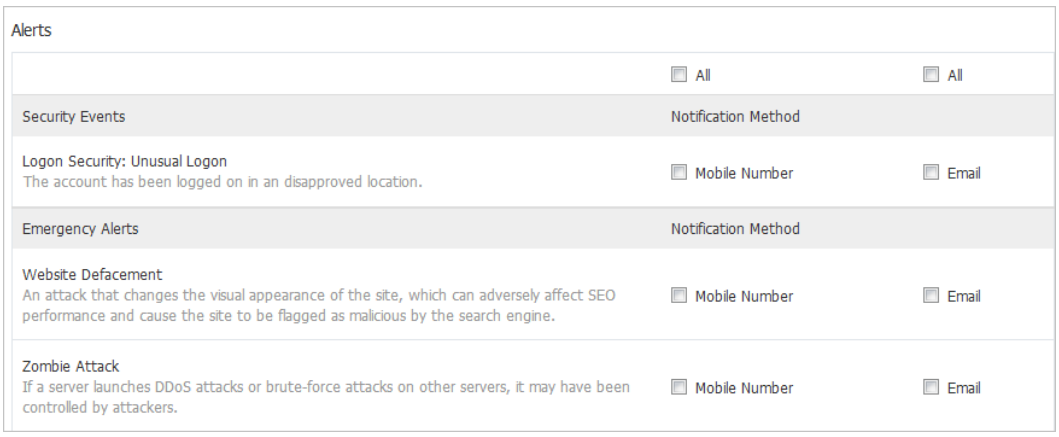
This topic describes how to configure the alert notification method for security events on tenants or platforms.

Context

In the **Alerts** section, security administrators can configure the alert notification method for security events. When a security event occurs, the system notifies the alert contacts by email, text message, or DingTalk. For more information about how to configure alert contacts, see Set alert recipients.

Alerts on tenants

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Alert Settings**. On the page that appears, click the **Tenant Alerts** tab.
3. In the **Alerts** section, select notification methods for each security event.



4. Click **Confirm**.

Alerts on the platform

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Alert Settings**. On the page that appears, click the **Platform Alerts** tab.
3. In the **Alerts** section, select notification methods for each security event.
4. Click **Confirm**.

27.1.9.8.3. Updates

27.1.9.8.3.1. Overview of the system updates feature

The system updates feature allows you to manually or automatically update the Apsara Stack Security and rule libraries for up-to-date protection.

The supported package import method depends on the Apsara Stack network environment.

- If Apsara Stack is connected to the Internet, you can choose **Automatically Download Update Packages**.
- If Apsara Stack is not connected to the Internet, you can choose **Manually Import Update Packages**.

The following table lists the update statuses of a rule library.

Update statuses of a rule library

Status	Description
To Be Updated	Indicates that a new version of the rule library is available for update.
Updating	Indicates that the rule library is being downloaded from Alibaba Cloud for update.
Updated	Indicates that the rule library has been updated.

Status	Description
Update Failed	Indicates that the rule library failed to be updated.

27.1.9.8.3.2. Enable automatic update check and update rule libraries

This topic describes how to enable automatic download of update packages and update rule libraries.

Context

If the Apsara Stack environment can connect to the Internet, you can enable automatic download of update packages to update the rule libraries.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Updates**.
3. Turn on **Auto Update Configuration** to enable automatic download of update packages.

   After this switch is turned on, the system automatically downloads update packages on a regular basis.
4. Update a rule library.

   You can update one or more rule libraries at a time.
   - Update multiple rule libraries at a time

     Click **Batch Update** in the upper-right corner to update all rule libraries.
   - Update a single rule library
     a. Click the tab of the rule type that you want to update. For example, click **Server Security**.
     b. In the **Actions** column, click **Update**.

27.1.9.8.3.3. Manually import an update package and update your service

This topic describes how to manually import an update package and update your service.

Prerequisites

The security administrator has obtained the offline update package.

Context

If the Apsara Stack environment cannot connect to the Internet, you can update a rule library after you import an offline update package.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Updates**.
3. Manually import an update package.
   i. Click **Import Update Package** next to **Manual Update** in the upper-left corner.

ii. In the **Import Update Package** dialog box, click **Browse** to select an offline update package that is downloaded to your on-premises server.

iii. Click **Confirm**.

4. Update a rule library.

   You can update one or more rule libraries at a time.

   - Update multiple rule libraries at a time

     Click **Batch Update** in the upper-right corner to update all rule libraries.

   - Update a single rule library

     a. Click the tab of the rule type that you want to update. For example, click **Server Security**.

     b. Click **Update** in the **Actions** column.

27.1.9.8.3.4. Roll back a rule library

This topic describes how to roll a rule library back to a previous version.

Context

If an error occurs with an updated rule library, you can roll the library back to a previous version to avoid service interruption.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Updates**.
3. Click the tab of the rule library that you want to roll back. Example: **Server Security**
4. In the Actions column for the rule library, choose **More > Roll Back**.
5. In the **Version Rollback** dialog box, click **Confirm**.

27.1.9.8.3.5. View the update history of a rule library

This topic describes how to view the update history of a rule library.

Context

You can view the update history of a rule library. If an error occurs with the latest version, you can locate the issue and roll back the rule library to an earlier version.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Updates**.
3. Click the tab of the specific rule library. Example: **Server Security**.
4. In the **Actions** column of a rule library, click **History**.

   On the **Previous Updates** page, you can view the update history of the rule library. Click **Details** to view the details of an update package.

27.1.9.8.4. Global configuration

27.1.9.8.4.1. Set CIDR blocks for traffic monitoring

Add a CIDR block for traffic monitoring

This topic describes how to add a Classless Inter-Domain Routing (CIDR) block for traffic monitoring. Network Traffic Monitoring System of Apsara Stack Security monitors the traffic of a specific CIDR block.

Context

CIDR blocks are configured for Network Traffic Monitoring System. Security administrators can change the CIDR blocks for monitoring based on business requirements. The settings of CIDR blocks apply only to a data center that is deployed in the region to which the specific CIDR block belongs.

> **Note**
>
> Changes to CIDR block settings immediately take effect without the intervention of security administrators.
>
> If you add the same CIDR block on the traffic collection CIDR block setting page and region setting page, make sure that you select the same region on both pages.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings**. On the page that appears, click the **Traffic Collection IP Range** tab.
3. Click **Add**.
4. In the **Add CIDR Block for Monitoring** dialog box, configure parameters.

   - **CIDR Block**: Enter a CIDR block for traffic monitoring.

     > **Note** Take note that the CIDR block that you entered must be valid and unique.

   - **Region**: Select the region of the data center.
5. Click **Confirm**.

Manage CIDR blocks for traffic monitoring

This topic describes how to modify or delete Classless Inter-Domain Routing (CIDR) blocks for traffic monitoring.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings**. On the page that appears, click the **Traffic Collection IP Range** tab.
3. Select a region, enter the CIDR block that you want to query, and then click **Search**.

   View the information about the CIDR block for traffic monitoring and the region in the search result.

4. In the **Actions** column, manage a CIDR block for traffic monitoring.
   - Modify the CIDR block for traffic monitoring

     Click **Modify** to modify the region of the CIDR block for traffic monitoring.
   - Delete the CIDR block for traffic monitoring

     Click **Delete** to delete the CIDR block for traffic monitoring.

27.1.9.8.4.2. Region settings

Add a CIDR block for a region

This topic describes how to add Classless Inter-Domain Routing (CIDR) blocks for regions that are detected and reported by using Server Guard.

Context

Region settings are used for region detection of the Server Guard agent. Server Guard servers automatically detect and match the regions of servers based on the IP address information that is reported by the Server Guard agent.

> **Note** You can change the region of a CIDR block. After the region is modified, you must also modify the region for all assets in the CIDR block on the Asset Overview page.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings**. On the page that appears, click the **Region** tab.
3. Click **Add**.
4. In the **Add CIDR Block** dialog box, configure parameters.

   - **CIDR Block**: Enter a CIDR block for the region.

     > **Note** Enter a valid CIDR block. You cannot enter a CIDR block that has been configured for the region.
   - **Region**: Select a region.
5. Click **Confirm**.

Manage CIDR blocks for a region

This topic describes how to modify or delete Classless Inter-Domain Routing (CIDR) blocks for a region.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings**. On the page that appears, click the **Region** tab.
3. Select a region, enter the CIDR block that you want to modify or delete, and then click **Search**.

   You can view the information about the CIDR block for the region in the search result.
4. In the **Actions** column, click Modify or Delete to manage the CIDR block for the region.
   - Modify the CIDR block for the region

     Click **Modify** to modify the CIDR block for the region.
   - Delete the CIDR block for the region

     Click **Delete** to delete the CIDR block for the region.

27.1.9.8.4.3. Configure whitelists

This topic describes how to configure the whitelist for the feature that blocks brute-force attacks in Server Guard and the following whitelists in Threat Detection Service (TDS). The whitelists contain IP addresses allowed by server brute-force attack blocking, IP addresses allowed by application attack blocking, and IP addresses allowed by web attack blocking.

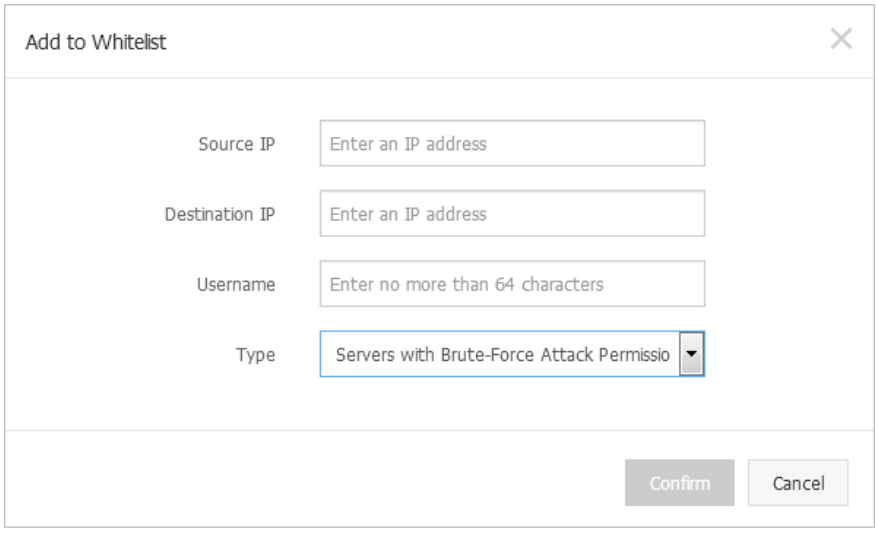
Context

If a normal request is regarded as an attack by the attack blocking feature of TDS or the unusual logon detection feature of Server Guard, you can add the source IP address of the request to a whitelist to avoid further false positives.

> **Note** Make sure that the IP addresses in the whitelist are trusted.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings**. On the page that appears, click the **Whitelist** tab.
3. Click **Add**.
4. In the **Add to Whitelist** dialog box, configure the parameters.



Parameter	Description
Source IP	Enter a source IP address or Classless Inter-Domain Routing (CIDR) block.
Username	Enter the name of the user who creates the whitelist.
Type	○ **Brute-Force Attack Blocking Whitelist**: Server Guard does not generate alerts for brute-force attacks or unusual logons from the IP addresses that are contained in this whitelist. ○ **BWAF Whitelist**: The attack blocking feature does not generate alerts for the web attacks from the IP addresses that are contained in this whitelist. ○ **Servers with Brute-Force Attack Permissions**: The attack blocking feature does not generate alerts for the brute-force attacks from the IP addresses that are contained in this whitelist. ○ **IPs with Application Attack Permissions**: The traffic from the IP addresses in this whitelist is not detected as suspicious application attack traffic.

5. Click **OK**.
   If you want to delete an existing whitelist, click **Delete** in the Actions column. In the **Delete Whitelist** message, click **Confirm**.

27.1.9.8.4.4. Configure attack blocking policies

This topic describes how to enable web attack blocking and brute-force attack blocking.

Context

The attack blocking features protect your servers against web attacks and brute-force attacks.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings > Region**.
3. Turn on or off the switches in the Actions column to enable or disable **Web Attack Blocking** or **Brute-Force Attack Blocking**.

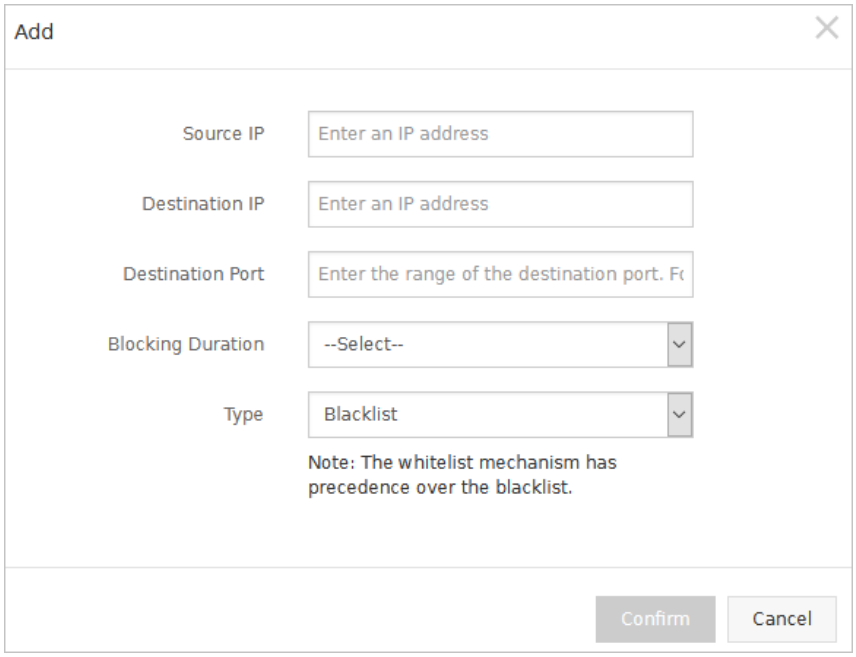
> **Note**
>
> In the Actions column, a red switch indicates a disabled feature and a green switch indicates an enabled feature.
>
> After you disable the blocking feature for an attack type, Apsara Stack Security Center generates only alerts for this type of attacks.

27.1.9.8.4.5. Block IP addresses

This topic describes how to manually block requests from a specific IP address based on traffic analysis results that are provided by Apsara Stack Security.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings**. On the page that appears, click the **Block IP Addresses** tab.
3. Click **Add**.
4. In the **Add** dialog box, configure the parameters.



Parameter	Description
Protocol Type	Specify the protocol type. Valid values: **IPv4** and **IPv6**.
Source IP	Enter the source IP address that you want to block.
Destination IP	Enter the destination IP address that you want to block.
Destination Port	Enter the destination port that corresponds to the specified destination IP address.
Blocking Duration	Select a time range during which you want to block requests. Valid values: **1 Day**, **7 Days**, and **30 Days**.
Type	Select the blocking mode. Valid values: **Whitelist** and **Blacklist**.
Description	Enter the reason for blocking.

5. Click **Confirm**.

27.1.9.8.4.6. Configure custom IP addresses and locations

Add custom IP addresses and locations

This topic describes how to add custom IP addresses and locations. You can customize internal IP addresses based on your network plan. After you configure the internal IP addresses, IP addresses from the public address library do not match the addresses outside China.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings**. On the page that appears, click the **Custom IP Location** tab.
3. Click **Add**.

   If you want to add multiple IP addresses and locations at a time, click **Batch Upload (.txt)**. This can be used to import multiple IP addresses and locations as a template.
4. In the **Add** dialog box, configure parameters.
5. Click **OK**.

Manage custom IP addresses and locations

This topic describes how to modify and delete custom IP addresses and locations.

Procedure

1.
2.
3. In the **Actions** column, manage custom IP addresses and locations.
   - To modify a custom IP address and a location:

     Click **Modify** to modify the custom IP address and location.
   - To delete a custom IP address and a location:

     Click **Delete** to delete the custom IP address and location.

27.1.9.8.5. System Monitoring

27.1.9.8.5.1. Configure CIDR blocks for traffic redirection in Cloud Firewall

Before you use Cloud Firewall, configure Classless Inter-Domain Routing (CIDR) blocks for traffic redirection.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > System Monitoring**.
3. On the **ICFW** tab in the **Service Monitoring** section of the **System Monitoring** page, click **Add** to the right of **Traffic Forwarding Configuration - Service CIDR Block**.
4. In the **Add** dialog box, configure **CIDR Block for Traffic Diversion** and **Type**.
5. Click **OK**.
   After you configure the CIDR blocks, you can view related information in **Traffic Forwarding Configuration - Service CIDR Block**, **Service Check Status**, and **Interface Status**.

27.1.9.8.6. Inspect services

This topic describes how to inspect services such as Cloud Firewall and Network Traffic Monitoring System in Apsara Stack Security Center. You can monitor the status and features of the services.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > System Monitoring**.
3. In the **System Inspection** section of the **Network Security** tab, inspect the services in the inspection list.

   To inspect a single service or multiple services at a time, perform the following operations:

   - Inspect multiple services at a time: In the **System Inspection** section, click **One-click Inspection** to inspect all services in the inspection list.
   - Inspect a single service: In the **System Inspection** section, click **Instant Inspection** in the **Actions** column of the service that you want to inspect.

   After you inspect the services, the status of the services changes to **Complete** in the **Inspection Status** column.
4. View the inspection results.

   You can view the following information about a service:

   - In the inspection list, view the service name, recent inspection time, number of inspection items, number of items whose status is normal, number of items whose status is abnormal, and inspection status.
   - Click **Details** in the **Actions** column of a service. In the **Inspection Details** dialog box, view the number of items whose status is normal, number of items whose status is abnormal, and the details of the items.
   - Click **Download** in the **Actions** column of a service. Download the inspection results to your computer as prompted for backup and reference.

27.1.9.8.7. Remote operations

27.1.9.8.7.1. Enable Remote O&M

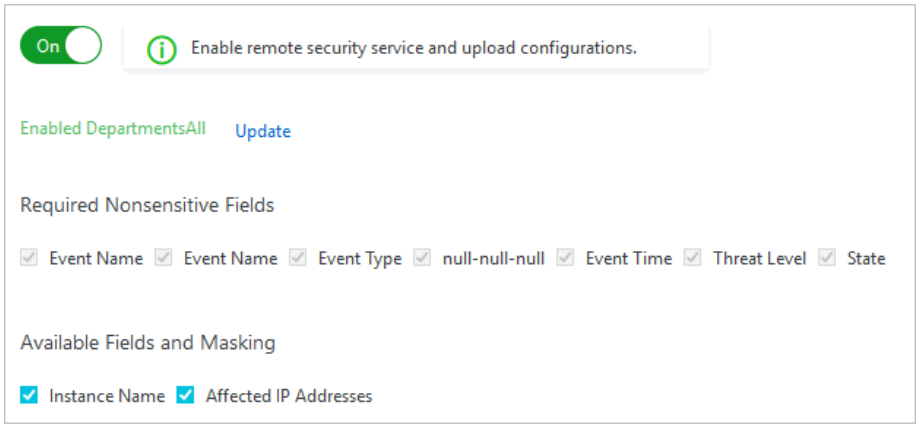
This topic describes how to enable remote operations.

Context

The **remote operations** function provides remote security operations and rules operations.

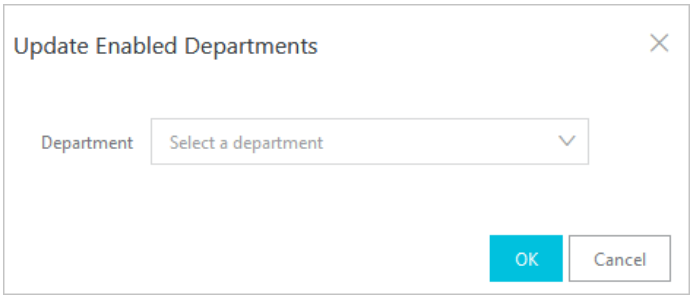
Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Remote O&M**.
3. On the **Remote O&M** page, turn on the switch to enable remote operations.



> **Note** The function is enabled if the switch turns green.

4. Select the departments for which you want to enable remote operations.
   i. Click **Update** on the right of **Enabled Departments**.
   ii. In the **Update Enabled Departments** dialog box, click the **Department** drop-down list to select the departments for which you want to enable remote operations.



   iii. Click **OK**.

> **Note** After you perform these operations, the security logs of **Enabled Departments** are encrypted and uploaded to Apsara Stack Security Center.

5. Select fields to encrypt and upload for remote operations.
   - If you select **Required Nonsensitive Fields**, the system encrypts and uploads the data.
   - If you select **Available Fields and Masking**, the system masks the data before encrypting and uploading it.

27.1.9.8.8. Account management

27.1.9.8.8.1. View and modify your Apsara Stack tenant account

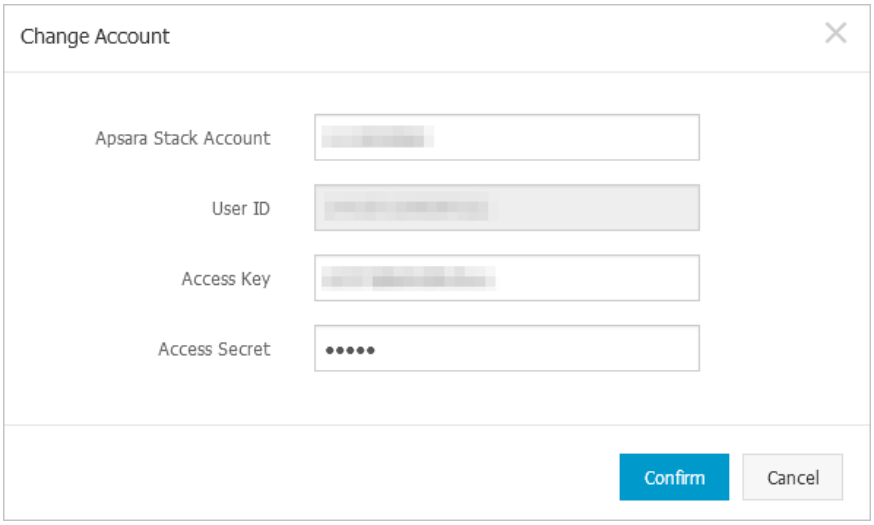
This topic describes how to view and modify the information about your Apsara Stack tenant account that is bound to Apsara Stack Security.

Context

Note All assets in Apsara Stack Security are bound to your Apsara Stack tenant account. You can modify the account information. Proceed with caution.

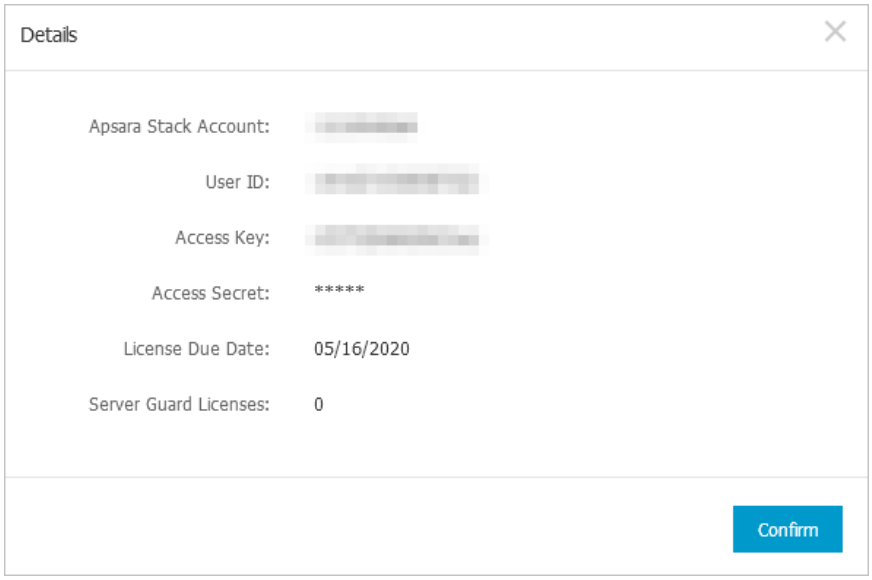
Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Accounts**. On the page that appears, click the **Apsara Stack Account** tab.
3. Modify the information about your Apsara Stack tenant account.
   i. In the Actions column, click **Modify**.
   ii. In the **Change Account** dialog box, modify the account information.



   iii. Click **Confirm**.
4. View the details of your Apsara Stack tenant account.
   i. In the Actions column, click **Details**.
   ii. View the account details.

   The details include the license expiration date and the number of Server Guard licenses. The details are obtained based on the user ID and the AccessKey pair.



27.1.9.8.8.2. Add an Alibaba Cloud account

This topic describes how to add an Alibaba Cloud account in Apsara Stack Security Center. This allows you to use features in a hybrid cloud.

Context

After you add an Alibaba Cloud account in Apsara Stack, you can manage the Anti-DDoS Pro, Anti-DDoS Premium, and Web Application Firewall (WAF) instances that belong to the Alibaba Cloud account in Apsara Stack. This allows you to use features in a hybrid cloud.

Procedure

1.
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Accounts**. On the page that appears, click the **Public Cloud Account** tab.
3. Click **Add**.
4. In the **Add Account** dialog box, enter the information about an Alibaba Cloud account and select Alibaba Cloud services to use.
   - Enter the **AccessKey ID** and **AccessKey secret** of the Alibaba Cloud account.
   - Select Alibaba Cloud services to use. Valid values:**Anti-DDoS Pro**, **Web Application Firewall**, and both.
5. Click **Confirm**.

Result

After an account is added, the account is displayed on the **Public Cloud Account** tab. To modify or delete an account, you can click **Modify** or **Delete** in the Actions column.

27.1.10. Optional security products

27.1.10.1. Anti-DDoS settings

27.1.10.1.1. Overview

In Distributed Denial of Service (DDoS) attacks, attackers exploit the client-server model to combine multiple computers into a platform that can launch attacks on one or more targets. This greatly increases the threat of attacks.

Common DDoS attack types include:

- **Network-layer attacks**: A typical example is UDP reflection attacks, such as NTP flood. These attacks use heavy traffic to congest the network of the victim, disabling proper responses to user requests.
- **Transport-layer attacks**: Typical examples include SYN flood and connection flood. These attacks consume a large number of connection resources of a server to cause denial of service.
- **Session-layer attacks**: A typical example is SSL flood. These attacks consume the SSL session resources of a server to cause denial of service.
- **Application-layer attacks**: Typical attack types include DNS flood, HTTP flood, and game zombie attacks. These attacks consume a large amount of application processing resources of a server to cause denial of service.

Apsara Stack Security can redirect, scrub, and re-inject attack traffic to protect your server against DDoS attacks and ensure normal business operations.

> **Note** Apsara Stack Security cannot scrub the traffic between internal networks.

27.1.10.1.2. View and configure DDoS mitigation policies

This topic describes how to view and configure distributed denial of service (DDoS) mitigation policies. Anti-DDoS provides default DDoS mitigation policies and DDoS traffic scrubbing policies.

Context

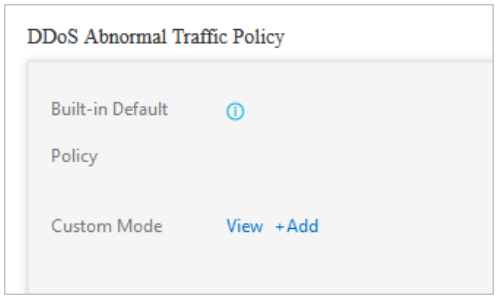
After an alert threshold of DDoS traffic for an IP address is set, an alert is triggered when traffic to the IP address reaches the threshold. The alert threshold for an IP address must be set based on the traffic volume. An abnormally large traffic volume indicates a possible DDoS attack. We recommend that you set an alert threshold to a value slightly higher than the peak traffic volume.

Apsara Stack Security supports a global alert threshold and alert thresholds for a specific Classless Inter-Domain Routing (CIDR) block or IP address.

- Global alert threshold: You cannot set a global alert threshold. It is automatically set when Apsara Stack Security is initialized.
- Alert threshold for a specific CIDR block: You can set an alert threshold for a specific CIDR block based on its traffic volume. CIDR block-specific alert thresholds allow you to control the traffic to each CIDR block.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Network Security > Policy Configuration > DDoS Defense Policy**.
3. View and customize DDoS mitigation policies.



Operation	Description
View the default policy	Move the pointer over the icon indicated by the exclamation point in the preceding figure to view the default DDoS mitigation policy.
Customize a policy	Click View to view CIDR block-specific policies, and click +Add to customize a DDoS mitigation policy for a CIDR block.

To customize a policy for a CIDR block, perform the following steps:

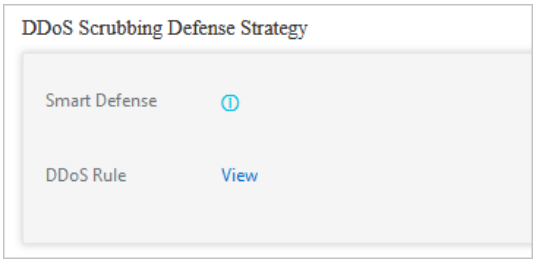
i. Click **+Add** next to **Custom Mode**.

ii. In the **Set Thresholds for Alerts** dialog box, configure the parameters.

Parameter	Description
CIDR Block	The CIDR block for which the alert thresholds are used.
Bandwidth Threshold	The alert threshold for bandwidth usage in a data center. When the sum of inbound and outbound traffic reaches this threshold, DDoS detection is triggered. Set this parameter to a value slightly higher than the peak traffic volume. We recommend that you set the value to 100 or higher. Unit: Mbit/s.
Packets Threshold	The alert threshold for the packet rate in a data center. When the sum of inbound and outbound packet rates reaches this threshold, DDoS detection is triggered. Set this parameter to a value slightly higher than the peak packet rate. We recommend that you set the value to 20000 or higher. Unit: packets per second (PPS).

iii. Click **OK**.

4. In the **DDoS Scrubbing Defense Strategy** section, click **View** to view DDoS traffic scrubbing policies.



27.1.10.1.3. View DDoS events

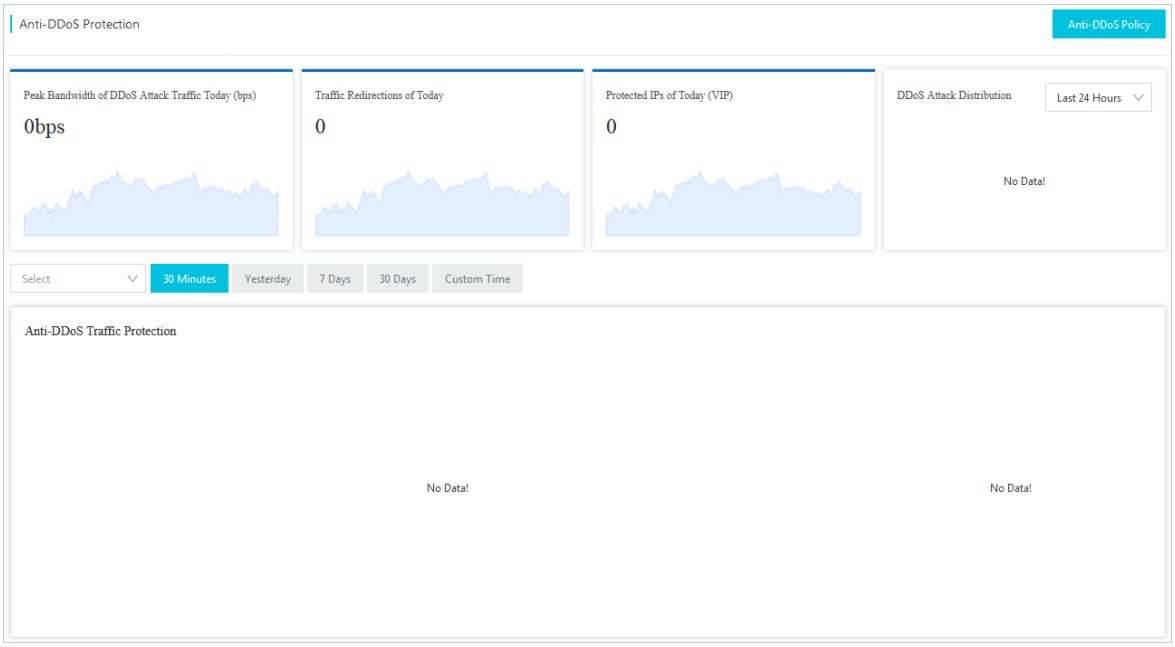
This topic describes how to view distributed denial of service (DDoS) events.

Context

During or after traffic scrubbing, Apsara Stack Security reports security events to Apsara Stack Security Center.

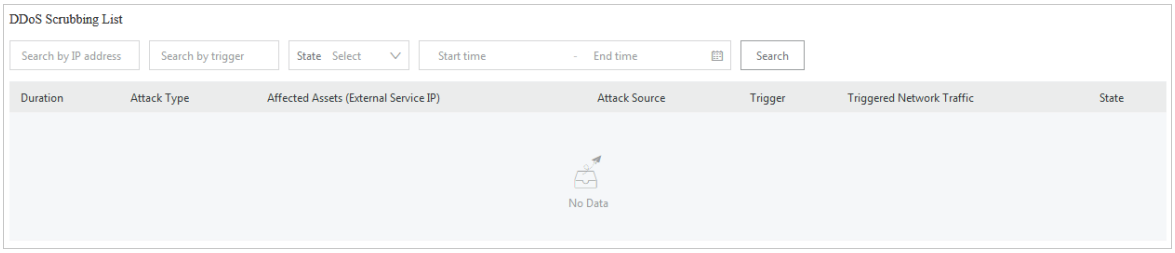
Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Network Security > DDoS Defense Policy**.
3. View anti-DDoS statistics.



4. (Optional)In the **DDoS Scrubbing List** section, specify search conditions and click **Search**.

> **Note** Skip this step if you need to view all traffic scrubbing events.



Search condition	Description
Search by IP address	The IP address that was under a DDoS attack.
Search by trigger	The metric that exceeds the configured alert threshold in the DDoS attack traffic.
State	○ Scrubbing: indicates that traffic scrubbing is in progress. ○ Scrubbing Complete: indicates that traffic scrubbing is complete.
Start time and End time	The start time and end time of DDoS traffic scrubbing.

5. In the DDoS Scrubbing List section, view details about DDoS traffic scrubbing events.

27.1.10.2. Cloud Firewall

27.1.10.2.1. Policy configuration

27.1.10.2.1.1. Synchronize assets for the Internet firewall

If new IP addresses are not in the IP address list of the Internet firewall, you can manually synchronize the IP address assets for the Internet firewall. This topic describes how to manually synchronize assets for the Internet firewall.

1.
2. In the left-side navigation pane, choose **Network Security > Policy Configuration > Firewall Switch Policy**.
3. Click the **Internet Firewall** tab. In the upper-right corner of the page, click **Update Assets**.
4. In the **Update Assets** message, click **OK**.
   After the assets are synchronized, the system updates the IP address list.

27.1.10.2.1.2. Create a VPC firewall

A virtual private cloud (VPC) firewall is a distributed firewall that can detect and control traffic between VPCs. Cloud Firewall can be used to analyze and control traffic between two VPCs only after a VPC firewall is created and enabled. This topic describes how to create a VPC firewall.

Context

A VPC firewall can be created only between two VPCs that are connected. VPCs can be connected by using Express Connect or Cloud Enterprise Network (CEN).

Procedure

1.
2. In the left-side navigation pane, click **Firewall Switch Policy**.
3. On the **Firewall Switches** page, click the **VPC-VPC** tab. Find the required instance and click **Create** in the Actions column.

Parameter	Description
Instance Name	Enter a name for the VPC firewall. We recommend that you enter a unique and informative name that indicates specific business to help identify the VPC firewall. The name can contain letters, digits, underscores (_), and hyphens (-). However, the name cannot contain only digits.
Route Table	When you create a VPC, the system automatically creates a default route table. You can add system routes to the route table to manage VPC traffic. VPC allows you to create multiple route tables. When you create a VPC firewall in the Cloud Firewall console, Cloud Firewall automatically reads your VPC route tables. Express Connect supports multiple route tables. If you create a VPC firewall for Express Connect, you can view multiple VPC route tables and can select specific route tables for protection.
Destination CIDR Block	After you select a route table from the Route Table drop-down list, the default destination Classless Inter-Domain Routing (CIDR) block of the route table appears in the Destination CIDR Block field. If you want to protect traffic destined for other CIDR blocks, you can manually modify the destination CIDR block. You can enter multiple CIDR blocks and separate them with commas (,).
Peer Route Table	Confirm the region and name of the peer VPC, and select the route table for protection.
Peer Destination CIDR Blocks	Confirm the destination CIDR block of the peer VPC.

Parameter	Description
Firewall Mode	○ Test: In this mode, the VPC firewall tests the health status of CIDR blocks or IP addresses to ensure normal links. ○ Active: In this mode, the VPC firewall redirects and protects traffic. ○ Bypass: In this mode, the VPC firewall does not redirect traffic. If a self-test or a health test fails, the VPC firewall automatically changes to the Bypass mode. When you create a VPC firewall, you can set Firewall Mode only to Test. After the firewall is created, you can change its mode to Active or Bypass. You cannot directly change the mode from Bypass to Active. You must change the mode from **Bypass** to **Test** first, and then to **Active**. In the **Health Test** section, enter 32-bit test IP addresses that belong to the local and peer VPCs.
IPS Mode	Select the working mode of the intrusion prevention system (IPS). Valid values: ○ Monitoring Mode: If Cloud Firewall detects malicious traffic, it monitors the traffic and sends alerts. ○ Traffic Control Mode: Cloud Firewall intercepts malicious traffic and blocks intrusion attempts.
IPS Capabilities	Select the intrusion prevention policies that you want to enable. Valid values: ○ Basic Policies: This feature provides basic intrusion prevention capabilities, such as protection against brute-force attacks and attacks that exploit command execution vulnerabilities. This feature also allows you to manage and control the connections from infected hosts to a command and control (C&C) server. ○ Virtual Patches: This feature defends against the most common and high-risk application vulnerabilities in real time.
Enable VPC Firewall	If you turn on Enable VPC Firewall, the VPC firewall is automatically enabled after it is created. If you do not require the VPC firewall to be automatically enabled, turn the switch off.

4. Click **Submit**.

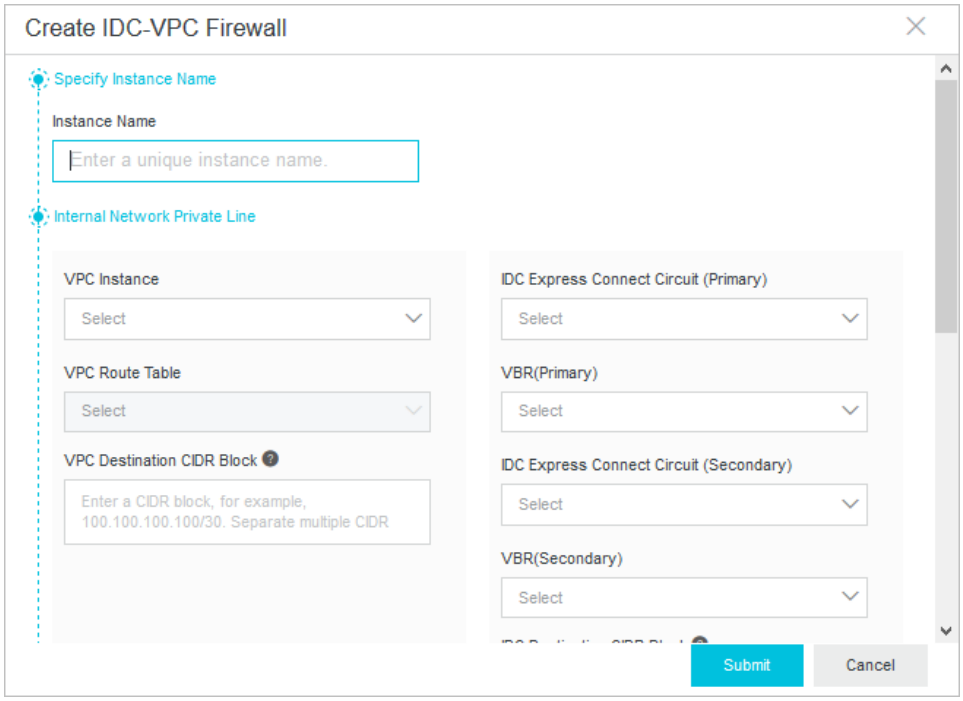
   If Firewall Status of the VPC firewall changes to **Enabled**, the VPC firewall takes effect.

27.1.10.2.1.3. Create an IDC-VPC firewall

An IDC-VPC firewall can detect the traffic between an on-premises data center and a virtual private cloud (VPC). Cloud Firewall allows you to control traffic through IDC-VPC firewalls. This topic describes how to create an IDC-VPC firewall.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, click **Firewall Switch Policy**.
3. On the **Firewalls** page, click the **IDC-VPC** tab. In the upper-right corner, click **Create**.



Parameter	Description
Instance Name	Enter a name for the VPC firewall. We recommend that you enter a unique name that indicates the specific business to make it easy to identify the VPC firewall. The name can contain letters, digits, underscores (_), and hyphens (-). However, the name cannot contain only digits.
VPC Instance	Select the ID of a VPC. The ID is the unique identifier of the VPC. Cloud Firewall automatically synchronizes the VPCs that are connected to the on-premises data center.
VPC Route Table	When you create a VPC, the system automatically creates a default route table. You can add system routes to the route table to manage VPC traffic. VPC allows you to create multiple route tables. When you create a VPC firewall in the Cloud Firewall console, Cloud Firewall automatically reads your VPC route tables. Express Connect supports multiple route tables. When you create a VPC firewall for an Express Connect, you can view multiple VPC route tables and can select the route tables that you want to protect.
VPC Destination CIDR Block	After you select a route table from the VPC Route Table drop-down list, the default destination Classless Inter-Domain Routing (CIDR) block of the route table is displayed in the Destination CIDR Block section. If you need to protect traffic to other CIDR blocks, you can manually modify destination CIDR blocks. You can add multiple CIDR blocks that are separated with commas (,).
IDC Express Connect Circuit (Primary)	Select a leased line ID that you create when you connect the on-premises data center to Apsara Stack. When the customer edge (CE) in the on-premises data center connects to the primary and secondary VSwitches, you must specify a primary leased line. The IDC-VPC firewall automatically synchronizes the primary leased line that you specify. You must specify this parameter when you create the IDC-VPC firewall.

Parameter	Description
VBR (Primary)	Select a virtual border router (VBR) that is bound to the primary leased line in the on-premises data center. The VBR facilitates communication between the VPC and the on-premises data center. The IDC-VPC firewall automatically synchronizes the VBR that you specify. You must specify this parameter when you create the IDC-VPC firewall.
IDC Express Connect Circuit (Secondary)	Select a leased line ID that you create when you connect the on-premises data center to Apsara Stack. You must specify a secondary leased line when the customer edge (CE) in the on-premises data center connects to the primary and secondary VSwitches. The value of this parameter cannot be the same as that of the IDC Express Connect Circuit (Primary) parameter. The IDC-VPC firewall automatically synchronizes the secondary leased line that you specify. You must specify this parameter when you create the IDC-VPC firewall.
VBR(Secondary)	Select a VBR that is bound to the secondary leased line in the on-premises data center. The VBR facilitates communication between the VPC and the on-premises data center. The IDC-VPC firewall automatically synchronizes the VBR that you specify. You must specify this parameter when you create the IDC-VPC firewall.
VPC Destination CIDR Block	The destination CIDR block of the peer VPC.
Firewall Mode	◦ Test: This mode is used to test the health status of the CIDR block or the IP address to ensure that the link is normal. ◦ Active: This mode is used to redirect and protect traffic. ◦ Bypass: The firewall does not redirect the traffic in this mode. If the self-test or the health test fails, the firewall is automatically changed to the Bypass mode. When you create a VPC firewall, you can set Firewall Mode only to Test. After the firewall is created, you can change its mode to Active or Bypass. You cannot directly change the mode from Bypass to Active. You must change the mode from **Bypass** to **Test** first and then to **Active**. In the **Health Test** section, enter a 32-bit test IP address that is created in the local VPC.
IPS Mode	Select the working mode of the intrusion prevention system (IPS). Valid values: ◦ Monitoring Mode: If Cloud Firewall detects malicious traffic, it monitors traffic and sends alerts. ◦ Traffic Control Mode: Cloud Firewall intercepts malicious traffic and blocks intrusion attempts.
IPS Capabilities	Select the intrusion prevention policies that you want to enable. Valid values: ◦ Basic Policies: This feature provides basic intrusion prevention capabilities such as protection against brute-force attacks and attacks that exploit command execution vulnerabilities. It also allows you to manage and control the connections from infected hosts to a command and control (C&C) server. ◦ Virtual Patches: This feature defends against the most common high-risk application vulnerabilities in real time.
Enable VPC Firewall	After you turn on Enable VPC Firewall, an IDC-VPC firewall is enabled automatically after it is created. If you do not require the IDC-VPC firewall to be automatically enabled after it is created, turn off this switch.

4. Click **Submit**.

   When the IDC-VPC firewall takes effect, Firewall Status of the IDC-VPC firewall changes to **Enabled**.

27.1.10.2.2. Access control

27.1.10.2.2.1. Manage address books

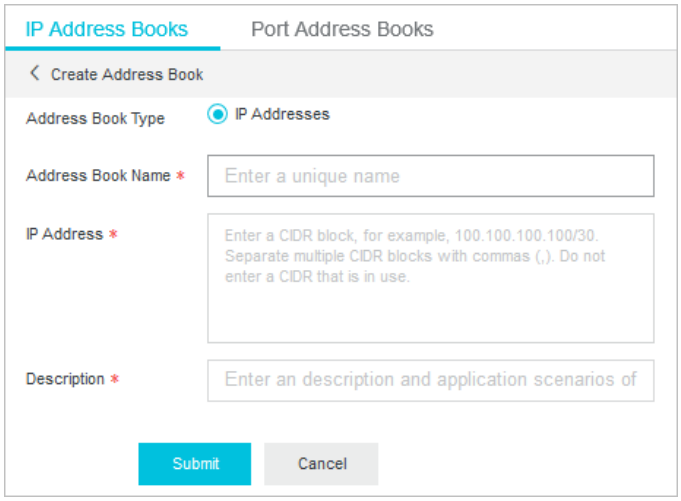
This topic describes how to create and manage IP address books and port address books. You can use address books to store one or more Classless Inter-Domain Routing (CIDR) blocks or ports.

Context

You can store frequently used IP addresses and ports in address books to facilitate configurations of the Internet firewall.

Create an IP address book

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Network Security > Access Control**.
3. On the Access Control page, click the **Internet Firewall** tab.
4. In the upper-right corner, click **Address Books**. In the dialog box that appears, click the **IP Address Books** tab.
5. Click **+Create Address Book** to specify parameters.



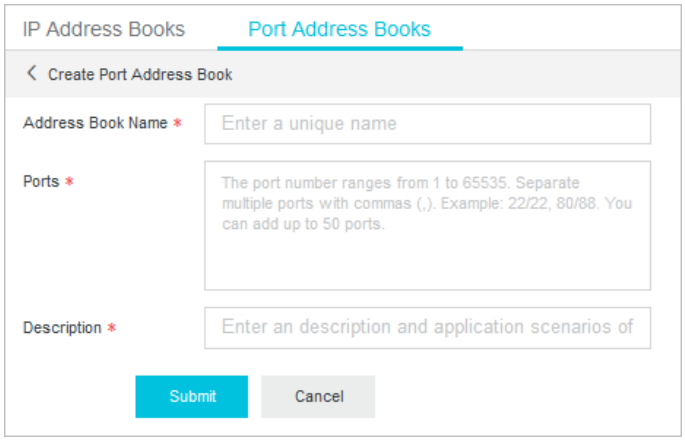
Parameter	Description
Address Book Type	Select the type of the address book. Set the value to **IP Addresses**.
Address Book Name	Specify the name of the address book. The name must be unique. The name can contain letters, digits, underscores (_), and hyphens (-). However, the name cannot contain only digits.
IP Address	Enter a CIDR block. Separate multiple CIDR blocks with commas (,).
Description	Enter the content and scenarios of the address book. The description must be 2 to 512 characters in length.

6. Click **Submit**.

After the address book is created, you can view the address book on the **IP Address Books** tab. You can click **Modify** or **Delete** to manage address books.

Create a port address book

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Network Security > Access Control**.
3. On the Access Control page, click the **Internet Firewall** tab.
4. In the upper-right corner, click **Address Books**. In the dialog box that appears, click the **Port Address Books** tab.
5. Click **+Create Address Book** to specify parameters.



Parameter	Description
Address Book Name	Specify the name of the address book. The name must be unique. The name can contain letters, digits, underscores (_), and hyphens (-). However, the name cannot contain only digits.
Ports	Enter a port number or port range. Separate multiple port numbers or ranges with commas (,).
Description	Enter the content and scenarios of the address book. The description must be 2 to 512 characters in length.

6. Click **Submit**.
After the address book is created, you can view the address book on the **Port Address Books** tab. You can click **Modify** or **Delete** to manage address books.

27.1.10.2.2.2. Configure access control policies on the Internet firewall

The access control feature allows you to configure access control policies on the Internet firewall. You can configure inbound and outbound policies on the Internet Firewall tab to forbid unauthorized access between the Internet and your servers.

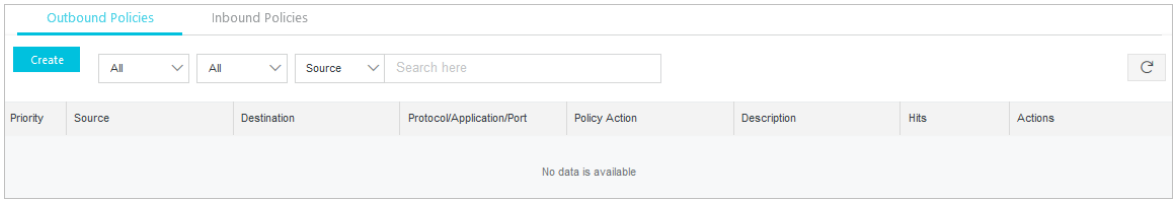
Context

The Internet firewall supports **Outbound Policies** and **Inbound Policies**.

In this topic, IP address books, port address books, and domain address books are used. For more information about how to create these address books, see Manage address books.

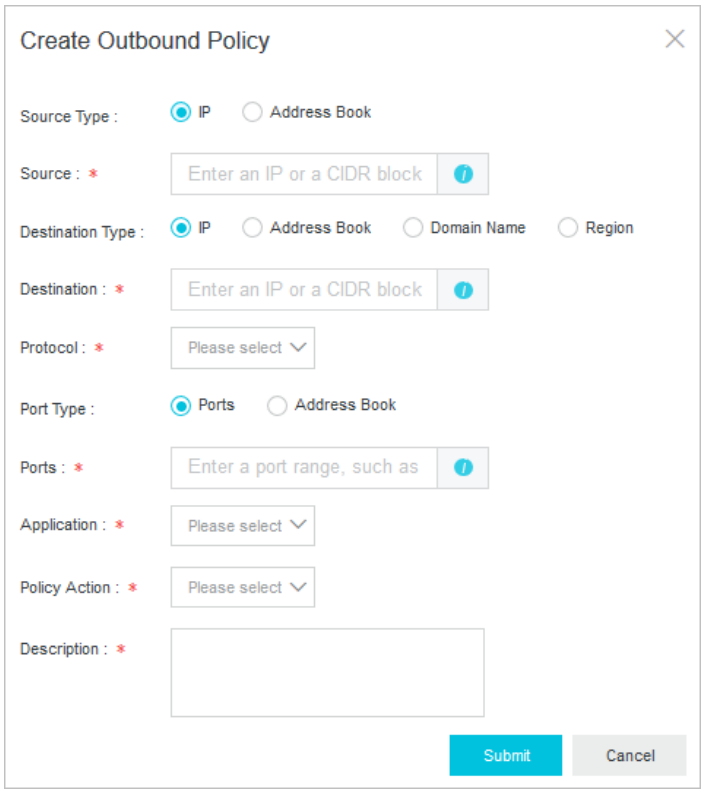
Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Network Security > Access Control**.
3. Click the **Internet Firewall** tab.
4. Click the **Outbound Policies** tab. You can also click the **Inbound Policies** tab if required.



   - **Outbound Policies** tab: You can configure control policies for traffic from your internal network to the Internet.
   - **Inbound Policies** tab: You can configure control policies for traffic from the Internet to your internal network.
5. Click **Create**.

   In the **Create Outbound Policy** dialog box, configure the parameters.



Parameter	Description

Parameter	Description
Source Type	The type of the traffic source. Valid values: **IP** and **Address Book**. ○ **IP**: If you select this option, enter only one CIDR block in the Source field. ○ **Address Book**: If you select this option, select a pre-configured IP address book for the Source field. An IP address book contains multiple CIDR blocks. This allows you to control traffic from multiple IP addresses.
Source	The source addresses that are allowed to access the Internet. ○ If you set **Source Type** to **IP**, enter a CIDR block. Example: 1.*.*.1/32. ○ If you set **Source Type** to **Address Book**, click **Select Address Book**. In the **Select Address Book as Source** dialog box, select an IP address book.
Destination Type	The type of the traffic destination. Valid values: **IP**, **Address Book**, **Domain Name**, and **Region**.
Destination	The destination addresses that can be accessed. You must set **Destination** to addresses on the Internet. ○ If you set **Destination Type** to **IP**, enter a CIDR block. Example: 1.*.*.1/32. ○ If you set **Destination Type** to **Address Book**, click **Select Address Book**. In the **Select Address Book as Destination** dialog box, select an IP address book. ○ If you set **Destination Type** to **Domain Name**, enter a domain name. Example: www.example.com. ○ If you set **Destination Type** to **Region**, select one or more destination regions from the drop-down list.
Protocol	The protocol of outbound traffic. Valid values: **TCP**, **UDP**, **ICMP**, and **ANY**. If you do not know which protocol is used, select ANY.
Port Type	The type of ports that are used for the selected protocol. Valid values: **Ports** and **Address Book**. ○ **Ports**: If you select this option, enter a port range in the Ports field. ○ **Address Book**: If you select this option, select a pre-configured port address book for the Ports field. A port address book contains multiple ports. This allows you to control traffic on multiple ports.
Ports	The ports on which you want to control traffic. Enter a port range or select a port address book based on how you set the Port Type parameter. ○ If you set **Port Type** to **Ports**, enter a port range. ○ If you set **Port Type** to **Address Book**, click **Select Address Book**. In the **Select Ports** dialog box, select a port address book.
Application	The application for the traffic.
Policy Action	The action on the traffic. ○ **Allow**: If outbound traffic meets the preceding conditions that you specify for the policy, the traffic is allowed. ○ **Monitor**: If outbound traffic meets the preceding conditions that you specify for the policy, the traffic is recorded and allowed. ○ **Deny**: If outbound traffic meets the preceding conditions that you specify for the policy, the traffic is blocked.

Parameter	Description
Description	The description of the policy. Enter an informative description to help identify the policy. The description can contain digits, letters, and underscores (_).

6. Click **Submit**.

Result

After the policy is created, it appears in the policy list. In the **Actions** column that corresponds to the policy, you can click **Modify**, **Delete**, **Insert**, or **Move** to manage the policy.

27.1.10.2.2.3. Create a policy group

This topic describes how to create a policy group for an internal firewall. You can configure access control policies to forbid unauthorized access between Elastic Compute Service (ECS) instances.

Context

An internal firewall is implemented by leveraging the security group module of ECS. The access control policies that you configure on the **Internal Firewall** tab are automatically synchronized to the security group module of ECS.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Network Security > Access Control**.
3. Click the **Internal Firewall** tab.
4. In the upper-right corner, click **Create Policy Group**.
5. In the **Create Policy Group** dialog box, configure parameters based on your business requirements.

Parameter	Description	Configuration method
Name	The name of the policy group. The name must be 2 to 128 characters in length.	Enter an informative name to help identify the policy group.
VPC	The virtual private cloud (VPC) to which the policy group is applied.	Select a VPC from the **VPC** drop-down list. **Note** You can select only one VPC.
Instance ID	The ID of the ECS instance in the selected VPC.	Select an ECS instance ID from the **Instance ID** drop-down list. **Note** You can select multiple instance IDs.
Description	The description of the policy group. The description must be 2 to 256 characters in length.	Enter an informative description to help identify the policy group.

Parameter	Description	Configuration method
Template	The template of the policy group.	Select a template from the **Template** drop-down list. Valid values: ○ **default-accept-login**: allows all inbound traffic on ports 22 and 3389. ○ **default-drop-all**: blocks all traffic in the policy group. ○ **default-accept-all**: allows all traffic in the policy group.

6. Click **Submit**.

27.1.10.2.2.4. Configure access control policies on an internal firewall

This topic describes how to view the policy groups of an internal firewall, configure access control policies in the policy groups, and synchronize the policies to the security group module of Elastic Compute Service (ECS).

Context

On the **Internal Firewall** tab, you can view custom policy groups and security groups that are synchronized from ECS.

For more information about how to create a custom policy group, see Create a policy group.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Network Security > Access Control**.
3. Click the **Internal Firewall** tab.
4. (Optional)Specify filter conditions and click **Search** to search for the required policy group.

   > **Note** If you want to view all policy groups, skip this step.

   The required policy group appears in the policy group list.
5. Find the required policy group and configure an access control policy in this group.
   i. In the **Actions** column, click **Configure Policy**.
   ii. On the **Policies** page, click **Create Policy**.

   > **Note** If you want to modify or delete an access control policy, you can perform the following steps: Click the **Inbound** or **Outbound** tab, find the policy, and then click **Modify** or **Delete** in the Actions column.

   iii. In the **Create Policy** dialog box, configure the parameters.

Parameter	Description
Network Type	The type of the network to which the policy is applied. Default value: **Internal**, which indicates that the policy is applied to an internal network.

Parameter	Description
Direction	The direction of traffic that is controlled by the policy. Valid values: ■ **Inbound**: Traffic is from other ECS instances to the specified ECS instance. ■ **Outbound**: Traffic is from the specified ECS instance to other ECS instances.
Policy Type	The action on traffic that is controlled by the policy. Valid values: ■ **Allow**: Traffic is allowed in the internal network. ■ **Deny**: Traffic is blocked in the internal network.
Protocol Type	The protocol of traffic that is controlled by the policy. Select a protocol from the **Protocol Type** drop-down list. Valid values: ■ **TCP** ■ **UDP** ■ **ICMP** ■ **ANY**: If you do not know which protocol is used, select **ANY**.
Port Range	The port range of traffic that is controlled by the policy. The traffic is destined for ports in this range. Enter a port range. Example: 22/22.
Priority	The priority of the policy. **Note** The priority number must be an integer from 1 to 100. Different policies can have the same priority. If an Allow policy and a Deny policy have the same priority, the **Deny** policy takes effect. If two **Allow** policies have the same priority, both policies take effect.
Source Type	The type of the traffic source. Valid values: ■ **CIDR Block**: The traffic source is a CIDR block. ■ **Policy Group**: The traffic source is an ECS instance in the policy group. **Note** If you create a policy in the security group module of ECS, you **cannot** set **Source Type** to **Policy Group**.
Source	The source of traffic that is controlled by the policy. Enter a CIDR block or select a policy group based on how you set the **Source Type** parameter. ■ If you set Source Type to **CIDR Block**, enter only one CIDR block. ■ If you set Source Type to **Policy Group**, select a policy group from the **Source** drop-down list. In this case, the traffic source is an ECS instance in the policy group. **Note** You can select only one policy group from the **Source** drop-down list.
Destination	The destination of traffic that is controlled by the policy. Valid values: ■ **All ECS Instances**: The policy is applied to traffic destined for your ECS instances. ■ **CIDR Block**: The policy is applied to traffic destined for the specified CIDR block.

Parameter	Description
Description	The description of the policy. Enter an informative description to help identify the policy. The description must be 2 to 256 characters in length.

iv. Click **Submit**.

6. Find the required policy group and click **Publish** in the **Actions** column to apply the policy and synchronize it to the security group module of ECS.

27.1.10.2.2.5. Configure access control policies on a VPC firewall

A virtual private cloud (VPC) firewall detects and controls the traffic between two VPCs. This topic describes how to configure access control policies on a VPC firewall.

Prerequisites

VPC firewalls are not automatically created. Before you configure access control policies for VPCs, you must create and enable a VPC firewall. For more information, see Create a VPC firewall.

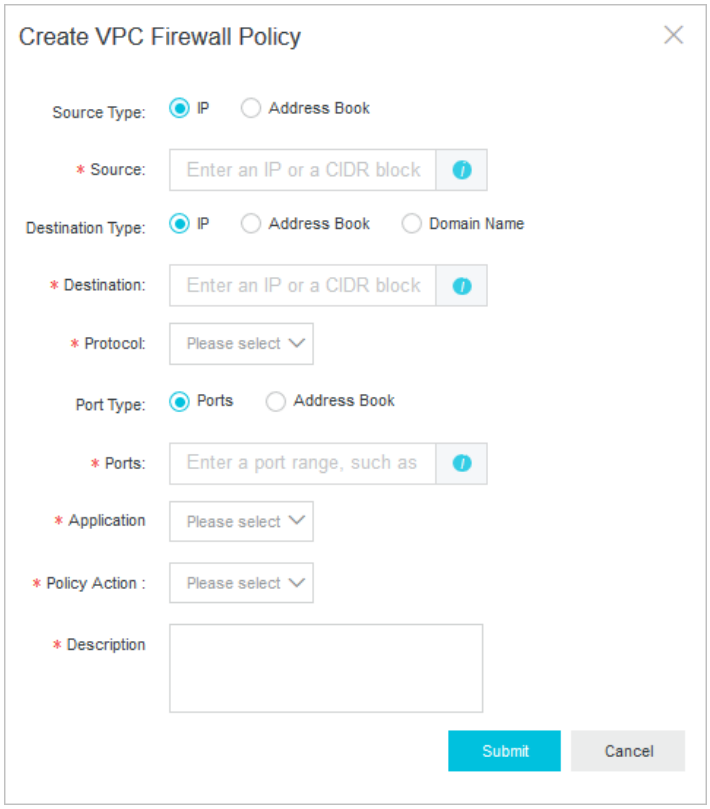
Context

Access control policies of a VPC firewall take effect only after you enable the VPC firewall.

By default, a VPC firewall allows all traffic. If you want to control traffic between VPCs, you can configure access control policies to block traffic from untrusted sources. You can also allow traffic from trusted sources and block traffic from all other sources.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Network Security > Access Control**.
3. Click the **VPC Firewall** tab.
4. Click **Create**.
5. In the **Create VPC Firewall Policy** dialog box, configure an access control policy.



Parameter	Description
Source Type	The type of the traffic source. Valid values: IP and Address Book. ○ **IP**: If you select this option, you must enter a CIDR block in the Source field. ○ **Address Book**: If you select this option, you must select a pre-configured address book for the Source field. You can add multiple CIDR blocks to an address book to simplify policy configuration.
Source	The source of the traffic. **Note** You can enter only one CIDR block. Example: 1.*.*.1/32. If you set Source Type to Address Book, select a pre-configured address book.
Destination Type	The type of the traffic destination. Valid values: ○ **IP**: If you select this option, you must enter an IP address in the Destination field. ○ **Address Book**: If you select this option, you must select an address book for the Destination field. ○ **Domain Name**: If you select this option, you must enter a domain name in the Destination field. Wildcard domain names are supported. Example: **.aliyun.com*. **Note** By default, if an HTTP header does not contain the Host field or an HTTPS request does not contain the Server Name Indication (SNI), Cloud Firewall allows the traffic.

Parameter	Description
Destination	The destination of the traffic. You can enter only one CIDR block. If you set Destination Type to Domain Name, enter a domain name. Wildcard domain names are supported. Example: **.aliyun.com*.
Protocol	The protocol of the traffic. Valid values: ○ **ANY**: all protocols ○ **TCP** ○ **UDP** ○ **ICMP**
Ports	The port range of the traffic. The value 0/0 indicates all ports. **Note** If you set Protocol to ICMP, the port configuration does not take effect. If you set Protocol to ANY, the port configuration does not take effect in controlling ICMP traffic.
Application	The application for the traffic. Valid values: ANY, HTTP, HTTPS, Memcache, MongoDB, MQTT, MySQL, RDP, Redis, SMTP, SMTPS, SSH, and VNC If you set Protocol to TCP, multiple applications are supported. If you set Protocol to another value, only ANY is supported. **Note** Cloud Firewall identifies applications based on packet characteristics, instead of port numbers. If Cloud Firewall fails to identify the application for a packet, it allows the packet.
Policy Action	The action on the traffic. Valid values: ○ **Allow**: If traffic meets the preceding conditions that you specify for the policy, the traffic is allowed. ○ **Deny**: If traffic meets the preceding conditions that you specify for the policy, the traffic is blocked. ○ **Monitor**: If traffic meets the preceding conditions that you specify for the policy, the traffic is recorded and allowed. After you observe the traffic for a period of time, you can change the policy action to Allow or Deny.
Description	The description of the policy. Enter an informative description to help identify the policy.

6. Click **Submit**.

27.1.10.2.2.6. Configure access control policies on an IDC-VPC firewall

Cloud Firewall detects and controls access traffic between Internet data centers (IDCs) and virtual private clouds (VPCs). This topic describes how to create and manage access control policies on an IDC-VPC firewall.

Prerequisites

IDC-VPC firewalls are not automatically created. Before you configure access control policies between an IDC and a VPC, you must create and enable an IDC-VPC firewall.

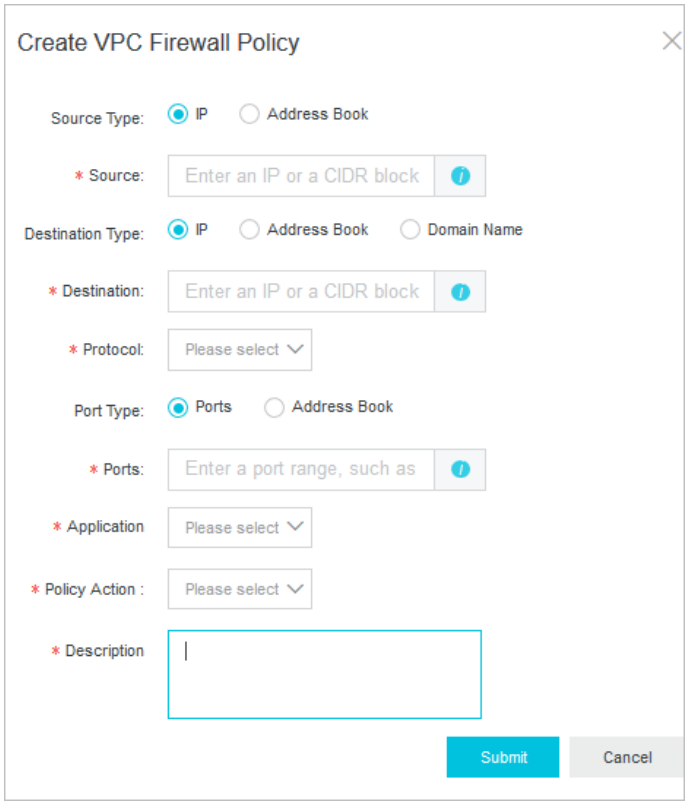
Access control policies of an IDC-VPC firewall take effect only after you enable the IDC-VPC firewall.

Context

By default, an IDC-VPC firewall allows all traffic. If you want to control traffic between an IDC and a VPC, you can configure access control policies to block traffic from untrusted sources. You can also allow traffic from trusted sources and block traffic from all other sources.

Procedure

1.
2. In the left-side navigation pane, choose **Network Security > Access Control**.
3. On the page that appears, click the **IDC-VPC Firewall** tab.
4. Click **Create**.
5. In the **Create IDC-VPC Firewall Policy** dialog box, configure an access control policy.



Parameter	Description
Source Type	The type of the traffic source. Valid values: ○ **IP**: If you select this option, you must enter an IP address or a Classless Inter-Domain Routing (CIDR) block in the Source field. ○ **Address Book**: If you select this option, you must select a pre-configured address book for the Source field. You can add multiple CIDR blocks to an address book to simplify policy configuration.

Parameter	Description
Source	The source of the traffic. **Note** You can enter only one CIDR block. Example: 1.*.*.1/32. If you set Source Type to Address Book, select a pre-configured address book.
Destination Type	The type of the traffic destination. Valid values: ○ **IP**: If you select this option, you must enter an IP address or a CIDR block in the Destination field. ○ **Address Book**: If you select this option, you must select an address book for the Destination field. ○ **Domain Name**: If you select this option, you must enter a domain name in the Destination field. Wildcard domain names are supported. Example: **.aliyun.com*. **Note** By default, if an HTTP header does not contain the Host field or an HTTPS request does not contain the Server Name Indication (SNI), Cloud Firewall allows the traffic.
Destination	The destination of the traffic. You can enter only one CIDR block. If you set Destination Type to Domain Name, enter a domain name. Wildcard domain names are supported. Example: **.aliyun.com*.
Protocol	The protocol of the traffic. Valid values: ○ **ANY**: all protocols ○ **TCP** ○ **UDP** ○ **ICMP**
Ports	The port range of the traffic. The value 0/0 indicates all ports. **Note** If you set Protocol to ICMP, the port configuration does not take effect. If you set Protocol to ANY, the port configuration does not take effect in controlling ICMP traffic.
Application	The application for the traffic. Valid values: ANY, HTTP, HTTPS, Memcache, MongoDB, MQTT, MySQL, RDP, Redis, SMTP, SMTPS, SSH, SSL, and VNC If you set Protocol to TCP, multiple applications are supported. If you set Protocol to another value, only ANY is supported. **Note** Cloud Firewall identifies applications based on packet characteristics, instead of port numbers. If Cloud Firewall fails to identify the application for a packet, it allows the packet.

Parameter	Description
Policy Action	The action on the traffic. Valid values: ○ **Allow**: If traffic meets the preceding conditions that you specify for the policy, the traffic is allowed. ○ **Deny**: If traffic meets the preceding conditions that you specify for the policy, the traffic is blocked. ○ **Monitor**: If traffic meets the preceding conditions that you specify for the policy, the traffic is recorded and allowed. After you observe the traffic for a period of time, you can change the policy action to Allow or Deny.
Description	The description of the policy. Enter an informative description to help identify the policy.

6. Click **Submit**.

27.1.10.2.3. Intrusion prevention

27.1.10.2.3.1. Configure intrusion prevention policies

Cloud Firewall uses a built-in threat detection engine to defend against intrusions and common cyber attacks. It provides virtual patches against vulnerabilities to intelligently block intrusion attempts.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Network Security > Policy Configuration > Intrusion Prevention Policies**.
3. Select a running mode for the threat engine.

Valid values: **Monitoring Mode** and **Traffic Control Mode**.

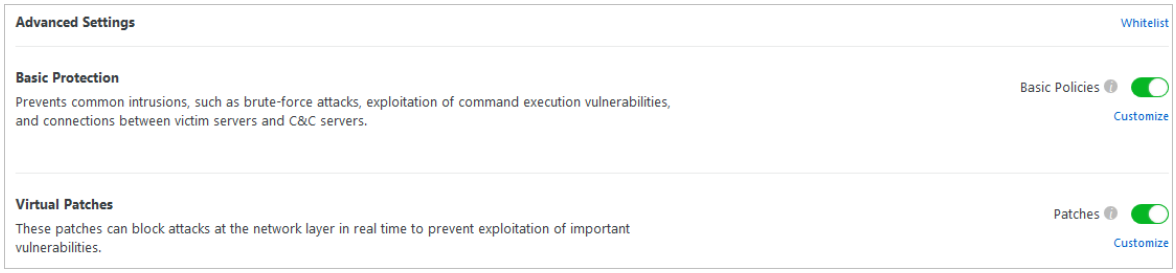
- **Monitoring Mode**: In monitoring mode, the system generates alerts on intrusions instead of blocking the malicious traffic.
- **Traffic Control Mode**: In traffic control mode, the system automatically blocks malicious traffic.

To configure **Traffic Control Mode**, perform the following steps:

i. Select **Traffic Control Mode**.

ii. In the dialog box that appears, click **OK**.

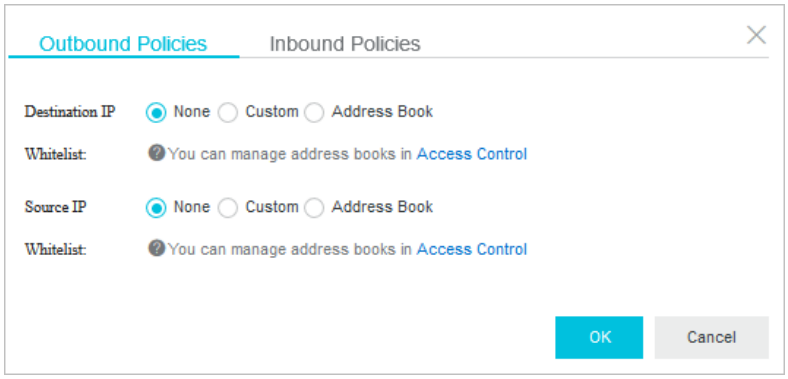
4. Configure the advanced features.



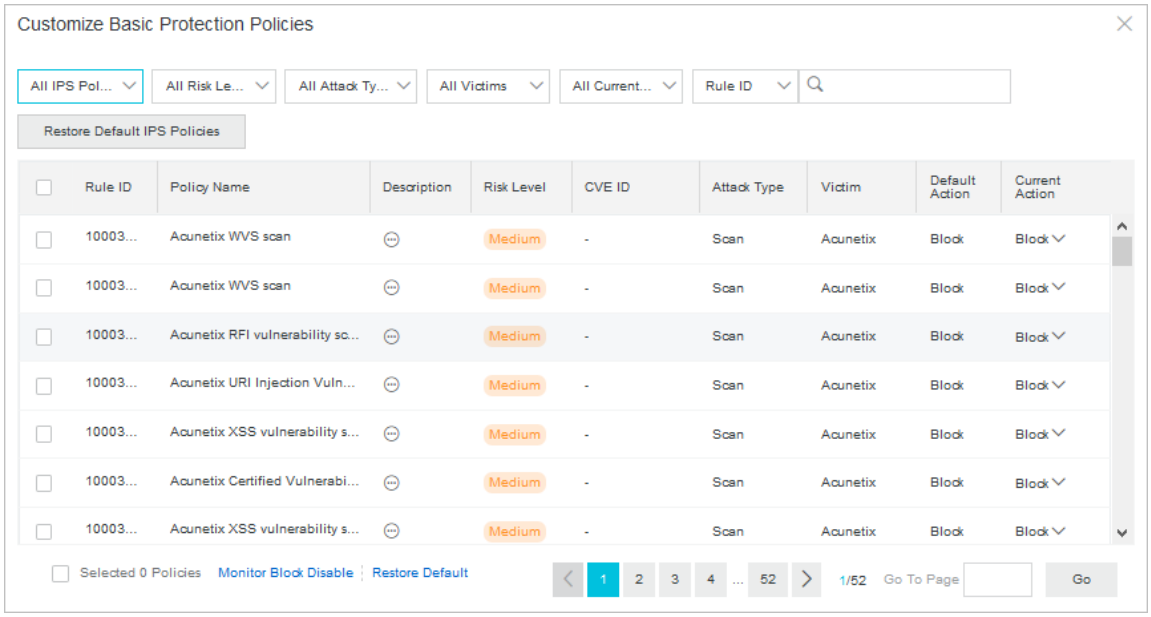
- Cloud Firewall allows traffic from IP addresses in the whitelist.

In the **Advanced Settings** section, click **Whitelist** to add trusted IP addresses to a whitelist.

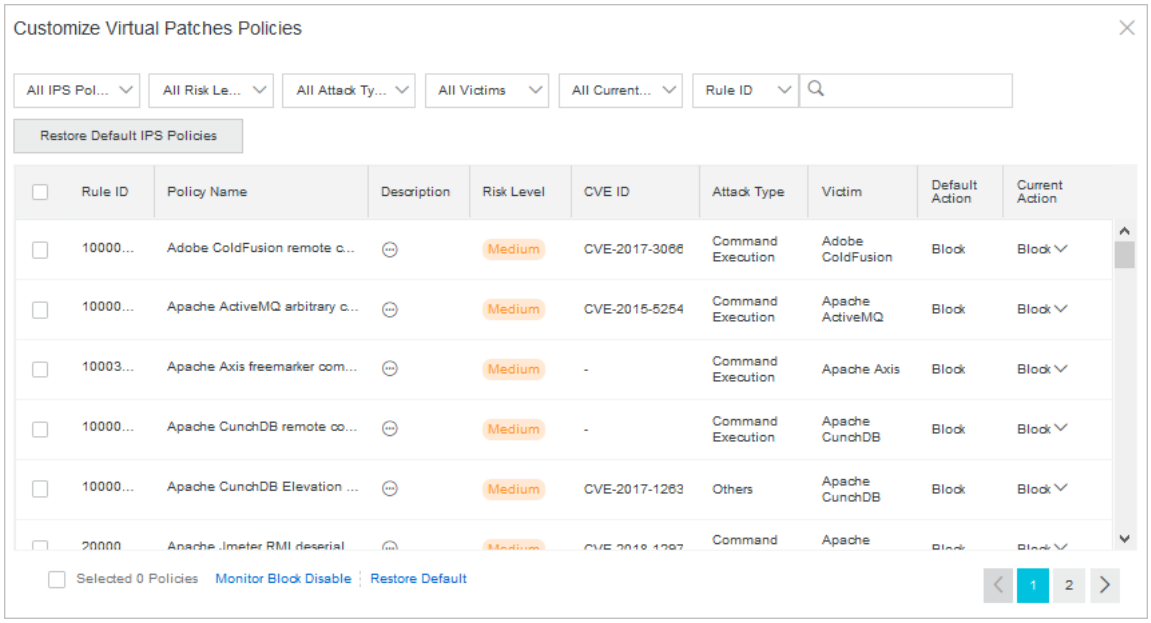
You can add the trusted source IP addresses, destination IP addresses, or address books of both inbound and outbound traffic to the whitelist.



- The basic protection feature defends your network against common intrusions, such as brute-force attacks and command execution vulnerabilities. The basic protection feature also manages connections from infected hosts to a command-and-control (C&C) server.
  a. Click **Basic Policies** to enable the basic protection feature.
  b. In the Basic Protection section, click **Customize**. In the **Customize Basic Protection Policies** dialog box, you can customize one or more basic protection policies.



- Virtual patches are installation-free. You can use them to defend against high-risk vulnerabilities.
  a. Click **Patches** to enable the virtual patch feature.
  b. In the **Virtual Patches** section, click **Customize**. In the **Customize Virtual Patches Policies** dialog box, configure one or more basic virtual patch policies.



27.1.10.2.3.2. View the traffic blocked by IPS

This topic describes how to view the details about Internet and VPC attacks that are blocked by the intrusion prevention system (IPS) feature of Cloud Firewall.

Procedure

1.
2. In the left-side navigation pane, choose **Network Security > Intrusion Prevention**.
3. Click the **Internet Traffic Blocking** and **VPC Traffic Blocking** tabs to view the details of the IPS-blocked traffic.
   - **Internet Traffic Blocking**

     On the **Internet Traffic Blocking** tab, you can view the blocking events of inbound and outbound traffic from the last one hour, one day, or seven days.

     The Internet Traffic Blocking tab contains the following sections:

     - **Defended Attacks**: displays the number and trend of attacks whose inbound or outbound traffic is blocked by IPS.
     - **Attack Types**: displays the distribution of attacks whose inbound or outbound traffic is blocked by IPS.
     - **Frequently Attacked Apps**: displays the applications whose inbound or outbound traffic is blocked by IPS.
     - **Frequently Attacked Destinations**: displays the traffic distribution of attack targets whose inbound or outbound traffic is blocked by IPS.
     - **Frequently Attacked Sources**: displays the traffic distribution of attack sources whose inbound or outbound traffic is blocked by IPS.
     - **Detailed Data**: displays the details of each traffic blocking event. The details include the risk level, number of times the event occurred, source IP address, and destination IP address.

       In the **Detailed Data** section, you can perform the following operations:

       - Specify a risk level, module, traffic direction, or time range to search for events.
       - Find a traffic blocking event and click **View Details** in the **Action** column to view the details. The details include the event description.

- **VPC Traffic Blocking**

  The **VPC Traffic Blocking** tab displays the suspicious traffic blocked by IPS between VPCs. You can view the details of a traffic blocking event over a specific time range. The details include the event name, risk level, and attack type.

  On the **VPC Traffic Blocking** tab, you can perform the following operations:

  - Specify a risk level, defense mode, attack type, or time range to search for events. After you specify the filter conditions, you must click **Search**.
  - Find a traffic blocking event and click **View Details** in the **Action** column to view the details. The details include the event description and defense mode.

27.1.10.2.4. View security groups

This topic describes how to view the details about and relationships between security groups and Elastic Compute Service (ECS) instances in a virtual private cloud (VPC).

1.
2. In the left-side navigation pane, choose **Network Security > Security Groups**.
3. On the **Visual Security Groups** page, view information about security groups and ECS instances in VPCs.

   Select a VPC and view the numbers of security groups, vSwitches, and ECS instances in the VPC.
4. Click the VPC of the security groups that you want to view. Then, you can view details about the security groups and ECS instances in the VPC.

   You can specify a time range to view the details. The time range can be one hour, one day, seven days, or one month.

   - View details about all security groups and ECS instances in the VPC.
     - On the right side of the **Visual Security Group** page, you can view the total numbers of security groups and ECS instances in the VPC, the number of first visits, and the list of security groups in the **Security Group** view.

       > **Note** **First Visits** indicates the total number of first visits between the ECS instances that have dependency relationships in a security group.

       By default, the **Visual Security Groups** page displays the icons of all security groups. In Security Group List, you can find a security group and click the icon in the **Actions** column to hide the icon of the security group on the Visual Security Groups page.
     - In the upper-right corner of the **Visual Security Groups** page, click **Data Details**. On the **Service Nodes** tab, you can view all the security groups in the VPC in the **Security Group** view.
   - View details about a single security group and the ECS instances in the security group.
     - On the **Visual Security Group** page, click the icon in the **Actions** column in Security Group List. On the **Service Nodes** tab, you can view details about the ECS instances in the security group in the **ECS** view.

- On the **Visual Security Groups** page, click the icon of a security group. On the right side of the page, you can view the total number of ECS instances in the security group, the number of dependency security groups, the number of dependent security groups, the number of first visits, and the lists of the two types of security groups in the **Security Group** view. Dependency security groups indicate the security groups on which the current security group depends, and dependent security groups indicate the security groups that depend on the current security group. The popover of the security group displays **data details**, including the following items:
  - ECS instance details: You can click the icon to view details about the ECS instances in the security group.
  - Dependency security groups: You can click the icon to view details about the dependency security groups.
  - Dependent security groups: You can click the icon to view details about the dependent security groups.
  - First visits: You can click the icon to view details about the security groups or ECS instances that have dependency relationships.
- View details about a single ECS instance.
  - On the **Visual Security Groups** page, double-click the icon of the security group that contains the ECS instance. Then, all ECS instances in the security group appear. On the right side of the page, you can view the total number of ECS instances in the security group, the numbers of dependency and dependent security groups, the number of first visits, and the ECS instance list in the **ECS** view.

    By default, the Visual Security Groups page displays the icons of all ECS instances in a security group. In ECS Instance List, you can find an ECS instance and click the icon in the **Actions** column to hide the icon of the ECS instance on the Visual Security Groups page.
  - On the **Visual Security Groups** page, click the icon of an ECS instance. On the right side of the page, you can view the total numbers of dependency and dependent ECS instances, the number of first visits, and the lists of the two types of security groups in the **ECS** view. The popover of the ECS instance displays **data details**, including the following items:
    - Dependency ECS instances: You can click the icon to view details about the dependency ECS instances.
    - Dependent ECS instances: You can click the icon to view details about the dependent ECS instances.
    - First visits: You can click the icon to view details about the ECS instances that have dependency relationships.

27.1.10.2.5. Log audit

27.1.10.2.5.1. View event logs

All traffic passing through Cloud Firewall is recorded on the Log Audit page. The logs are classified into traffic logs and event logs. You can use the logs to audit your network traffic in real time and take actions accordingly. This topic describes how to view event logs.

Procedure

1. Log on to Apsara Stack Security Center.

2. In the left-side navigation pane, choose **Network Security > Log Audit > Event Log**.
3. On the **Event Logs** tab, click **Internet Firewall**, **VPC-VPC Firewall** or **IDC-VPC Firewall**.
4. Specify the search conditions and click **Search**.

> **Note** If you want to view all event logs, skip this step.

Search condition	Description
Source IP	Specify the source IP address of the event.
Destination IP	Specify the destination IP address of the event.
Type	Select the event type
Action	Select the event action. Valid values: *All*, *Monitor*, and *Discard*.
Time	Set the time range.

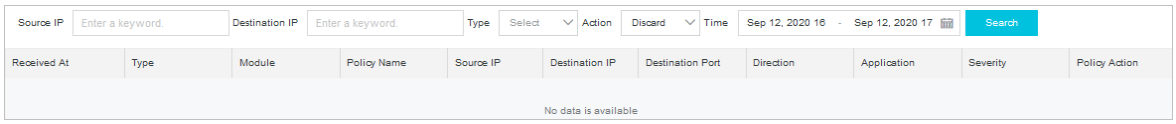
5. The event logs record the information of event. This includes the event detection time, threat type, traffic direction (inbound or outbound), source IP address, destination IP address, application type, severity, and policy action.

27.1.10.2.5.2. View traffic logs

All traffic passing through Cloud Firewall is recorded on the Log Audit page. The logs are classified into traffic logs and event logs. You can use the logs to audit your network traffic in real time and take actions accordingly. This topic describes how to view traffic logs.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Network Security > Log Audit > Traffic Log**.
3. On the **Event Logs** tab of the page that appears, click **Internet Firewall**, **VPC-VPC Firewall**, or **IDC-VPC Firewall**.
4. Specify the search condition and click **Search**.



> **Note** If you want to view all traffic logs, skip this step.

Search condition	Description
Source IP	Specify the source IP address of the traffic.
Destination IP	Specify the destination IP address of the traffic.
Application	Select the application type.
Time	Set the time range.

To enable the advanced features, perform the following steps:

- **Show Advanced Search**

Click **Show Advanced Search** to configure more search conditions.

- **List Configuration**

  Click **List Configuration** to select items for the traffic list.

5. The traffic logs record the information of access traffic. This includes the start time and end time of the access traffic, traffic direction (inbound or outbound), source IP address, destination IP address, application type, supported protocol, bytes, and packets.

27.1.10.3. Sensitive Data Discovery and Protection

27.1.10.3.1. Grant access permissions

Before you use Sensitive Data Discovery and Protection (SDDP), you must grant access permissions to SDDP. This topic describes how to authorize SDDP to access the data of your department.

Prerequisites

The name and AccessKey pair of your department are obtained before you grant access permissions on the department. For more information, see the "Obtain the AccessKey pair of an organization" topic in *Apsara Uni-manager Management Console User Guide*. To find the topic, choose **Enterprise > Organizations > Obtain the AccessKey pair of an organization**.

Context

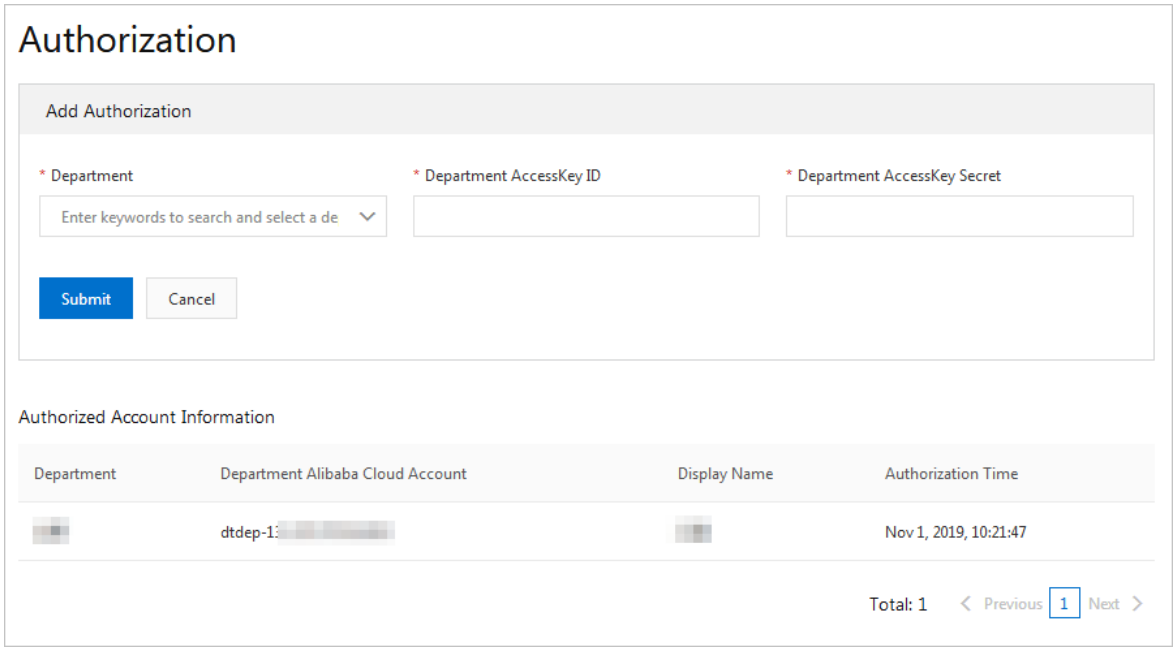
Before you use SDDP, you must complete the following operations:

- Authorize SDDP to access the data of your department.
- Authorize SDDP to access the data of Apsara Stack services of your department. The services include MaxCompute, Object Storage Service (OSS), and Tablestore.

Procedure

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Authorization**.

> **Note** If SDDP is not authorized to access the data of your department, the **Use authorization settings** page appears. You must configure the parameters on this page.



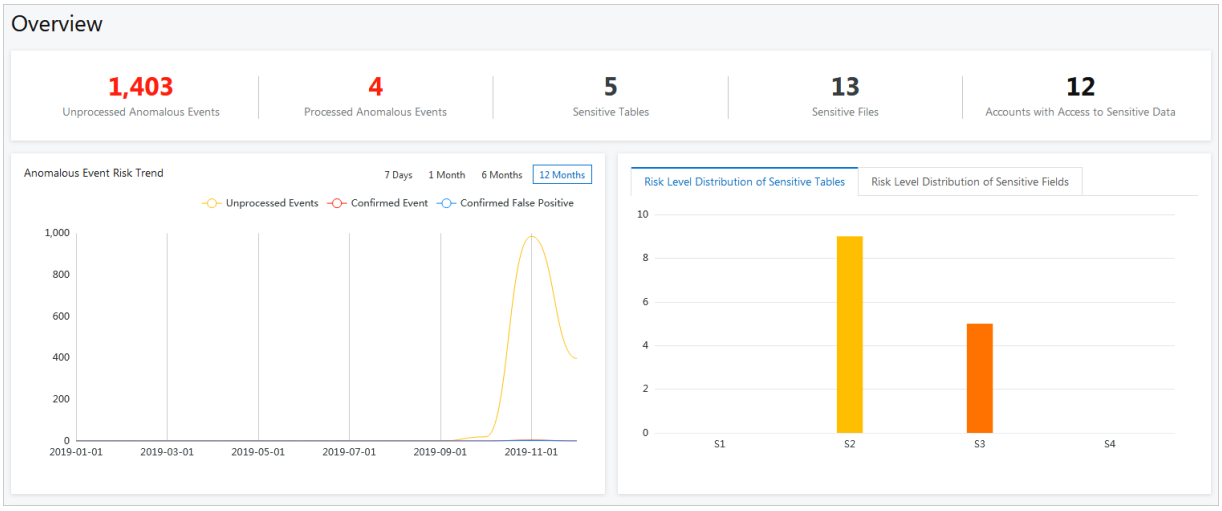
3. In the **Add Authorization** section, authorize SDDP to access the data of your department.
   i. In the **Department** drop-down list, enter a keyword and select the department.
   ii. Enter **Department AccessKey ID** and **Department AccessKey Secret**.
   iii. Click **Submit**.
4. In the **Authorized Account Information** section, view the departments that SDDP is authorized to access.

27.1.10.3.2. Overview

This topic describes the Overview page of Sensitive Data Discovery and Protection (SDDP). The Overview page displays the overall security status of data protected by SDDP. This allows security administrators to take an overview of sensitive data and take countermeasures in time.

SDDP can detect sensitive data in your data assets based on specific detection rules and track the use of sensitive data. SDDP also provides a data overview for you to obtain the security status of your data assets in real time.

If you want to view the overall security status of the sensitive data, log on to Apsara Stack Security Center. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Overview**.



- **Overview**: displays the overall information about sensitive data, including the number of **unhandled anomalous events**, the number of **handled anomalous events**, the number of **sensitive tables**, the

number of **sensitive objects**, and the **accounts that are used to access sensitive data**.

- **Abnormal Event Risk Trend**: displays the trends of different events in a line chart. You can select **7 Days**, **1 Month**, **6 Months**, or **12 Months** to view the trends of different events, such as **Unprocessed Events**, **Confirmed Event**, and **Confirmed False Positive**.
- **Sensitive table risk level distribution**: displays the distribution of sensitive tables at each sensitivity level, including L3 (highly sensitive), L2 (moderately sensitive), L1 (low sensitive), and N/A (non-sensitive).
- **Sensitive field risk level distribution**: displays the distribution of sensitive fields at each sensitivity level, including L3 (highly sensitive), L2 (moderately sensitive), L1 (low sensitive), and N/A (non-sensitive).
- **Data flow situation**:
  - Displays the dynamic statistics on core data flows in DataHub and Data Integration.
  - Provides a data flowchart. This flowchart dynamically shows data flows and abnormal output. You can click an anomalous event in the flowchart to go to the **Abnormal data flow** page.

    Monitors the data links among different entities, such as data storage services, data transmission services, the data flow processing service, external databases, and external files. The data storage services include MaxCompute, AnalyticDB for MySQL, Object Storage Service (OSS), and Tablestore. The data transmission services include DataHub and Data Integration. The data flow processing service includes Blink.

27.1.10.3.3. Data asset authorization

27.1.10.3.3.1. Authorize SDDP to access data assets

Sensitive Data Discovery and Protection (SDDP) must be authorized to access your data assets before it can detect sensitive data in the data assets. Supported data assets include Object Storage Service (OSS) buckets, ApsaraDB RDS instances, PolarDB-X databases, Tablestore instances, self-managed databases hosted on Elastic Compute Service (ECS) instances, MaxCompute projects, AnalyticDB for MySQL clusters, ApsaraDB for OceanBase clusters, and AnalyticDB for PostgreSQL instances. This topic describes how to authorize SDDP to access your data assets.

Context

SDDP can access and scan specific data assets to detect and mask sensitive data only after you grant the required permissions to SDDP.

> **Notice** 已开启授权的OSS Bucket（OSS文件桶）会消耗您的OSS存储容量，已开启授权的数据库或项目会消耗您的数据库和项目数。只有在OSS存储容量、数据库和项目数量充足时，您才可以成功进行相应授权操作。您可以在**云上托管**页面查看剩余的OSS存储容量、数据库和项目数。

For more information about how to authorize SDDP to access supported data assets, see the following sections:

- Authorize SDDP to access OSS buckets
- Authorize SDDP to access ApsaraDB RDS instances
- Authorize SDDP to access PolarDB-X databases
- Authorize SDDP to access Tablestore instances
- Authorize SDDP to access self-managed databases hosted on ECS instances
- Authorize SDDP to access MaxCompute projects
- Authorize SDDP to access AnalyticDB for MySQL clusters
- Authorize SDDP to access AnalyticDB for PostgreSQL instances
- Authorize SDDP to access ApsaraDB for OceanBase clusters

Authorize SDDP to access OSS buckets

1. Log on to Apsara Stack Security Center.

2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data protection authorization > Data Asset authorization**.
3. On the **OSS** tab, grant the required permissions on instances or buckets.
   - If you want to grant permissions on a single instance or bucket, turn on or off the switches in the **Identify permissions**, **Desensitization permissions**, **OCR Authority**, and **Audit permissions** columns of the instance or bucket. Then, configure the **Sensitive data sampling** and **Audit log archiving** parameters.

Parameter	Description
Identify permissions	The permissions to detect sensitive data in selected data assets.
Desensitization permissions	The permissions to mask sensitive data in selected data assets.
OCR Authority	The permissions to detect sensitive data in text on images.
Audit permissions	The permissions to audit data in selected data assets.
Sensitive data sampling	The number of sensitive data samples to reserve after SDDP detects sensitive data. If you turn on the switch in **Identify permissions**, you must also specify this parameter. If SDDP detects sensitive data in a data asset, it collects samples of the detected data. You can use the sensitive data samples for further analysis. Valid values: **0**, **5**, and **10**.
Audit log archiving	The duration to retain the audit logs of selected data assets. If you turn on the switch in **Audit permissions**, you must also specify this parameter. Valid values: **30 days**, **90 days**, and **180 days**. **Note** You do not need to activate Log Service to archive the audit logs generated by SDDP.

   - If you want to grant permissions on multiple instances or buckets at the same time, perform the following steps:
     a. Select the required instances or buckets and click **Batch operation**.
     b. In the **Batch processing for selected assets** dialog box, turn on or off the switches of detection, audit, masking, and OCR permissions, and configure the parameters that remain.
     c. Click **Confirm**.

   After SDDP is authorized, SDDP scans the OSS buckets to detect sensitive data. If SDDP scans an OSS bucket for the first time, SDDP automatically performs a full scan.

   In the list of OSS buckets on which SDDP has access permissions, you can modify or revoke permissions on the OSS buckets. If you revoke permissions on an OSS bucket, SDDP no longer scans the OSS bucket.

   Note SDDP scans only the accessible OSS buckets and analyzes the risks of sensitive data detected in these OSS buckets.

Authorize SDDP to access ApsaraDB RDS instances

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data protection authorization > Data Asset authorization**.

3. On the **Cloud hosting** page, click the **RDS** tab.
4. On the **RDS** tab, click **Unauthorized**.
5. Find the instance that you want SDDP to access and enter the required database username and its password in the **Username** and **Password** columns.

   You can also click **Batch password import** in the upper-right corner and upload an authorization file to import the logon information of multiple databases at the same time. For more information, see Import the logon information for multiple data assets at the same time.

   > **Notice** If the username or password is not correct, the authorization fails. Make sure that the information you enter is correct.

6. Select the databases that you want SDDP to access and click **Batch operation**.

   You can also click **Authorization** in the Actions column of an instance to grant all its permissions.

7. In the Batch processing for selected assets dialog box, turn on or off the switches of detection, audit, and masking permissions, and configure the parameters that remain.

Parameter	Description
Identify permissions	The permissions to detect sensitive data in selected data assets.
Audit permissions	The permissions to audit data in selected data assets. SDDP allows you to collect audit logs that cover the data generation, update, and use of your data assets. The log information includes the audit rule that is triggered for a data asset, the type of the data asset, the type of the operation that triggers the audit rule, and the operator account. SDDP安全审计功能的使用，请参见Create an audit rule。
Desensitization permissions	The permissions to mask sensitive data in selected data assets.
Sensitive data sampling	The number of sensitive data samples to reserve after SDDP detects sensitive data. If you turn on the switch in **Identify permissions**, you must also specify this parameter. If SDDP detects sensitive data in a data asset, it collects samples of the detected data. You can use the sensitive data samples for further analysis. Valid values: **0**, **5**, and **10**.
Audit log archiving	The duration to retain the audit logs of selected data assets. If you turn on the switch in **Audit permissions**, you must also specify this parameter. Valid values: **30 days**, **90 days**, and **180 days**.

8. Click **Confirm**.

   > **Note** If the authorization fails, check whether the username and password are correct.

   After SDDP is authorized, SDDP scans the databases to detect sensitive data.

   In the list of databases on which SDDP has access permissions, you can modify or revoke permissions on the databases. You can modify only the username and password of a valid database account. If you revoke permissions on a database, SDDP no longer scans the database.

Authorize SDDP to access PolarDB-X databases

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data protection authorization > Data Asset authorization**.
3. On the **Cloud hosting** page, click the **DRDS** tab.
4. On the **DRDS** tab, click **Unauthorized**.
5. Find the instance that you want SDDP to access and enter the required database username and its password in the **Username** and **Password** columns.

   You can also click **Batch password import** in the upper-right corner and upload an authorization file to import the logon information of multiple databases at the same time. For more information, see Import the logon information for multiple data assets at the same time.

   > **Notice** If the username or password is not correct, the authorization fails. Make sure that the information you enter is correct.

6. Select the databases that you want SDDP to access and click **Batch operation**.

   You can also click **Authorization** in the Actions column of an instance to grant all its permissions.

7. In the Batch processing for selected assets dialog box, turn on or off the switches of detection, audit, and masking permissions, and configure the parameters that remain.

Parameter	Description
Identify permissions	The permissions to detect sensitive data in selected data assets.
Audit permissions	The permissions to audit data in selected data assets. SDDP allows you to collect audit logs that cover the data generation, update, and use of your data assets. The log information includes the audit rule that is triggered for a data asset, the type of the data asset, the type of the operation that triggers the audit rule, and the operator account. SDDP安全审计功能的使用，请参见Create an audit rule。
Desensitization permissions	The permissions to mask sensitive data in selected data assets.
Sensitive data sampling	The number of sensitive data samples to reserve after SDDP detects sensitive data. If you turn on the switch in **Identify permissions**, you must also specify this parameter. If SDDP detects sensitive data in a data asset, it collects samples of the detected data. You can use the sensitive data samples for further analysis. Valid values: **0**, **5**, and **10**.
Audit log archiving	The duration to retain the audit logs of selected data assets. If you turn on the switch in **Audit permissions**, you must also specify this parameter. Valid values: **30 days**, **90 days**, and **180 days**.

8. Click **Confirm**.

   > **Note** If the authorization fails, check whether the username and password are correct.

   After SDDP is authorized, SDDP scans the databases to detect sensitive data.

In the list of databases on which SDDP has access permissions, you can modify or revoke permissions on the databases. You can modify only the username and password of a valid database account. If you revoke permissions on a database, SDDP no longer scans the database.

Authorize SDDP to access Tablestore instances

You can authorize SDDP to access one or more Tablestore instances.

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data protection authorization > Data Asset authorization**.
3. On the **Cloud hosting** page, click the **OTS** tab.
4. On the **OST** tab, grant the required permissions on instances or buckets.
   - If you want to grant permissions on a single instance or bucket, turn on or off the switches in the **Identify permissions**, **Desensitization permissions**, and **Audit permissions** columns of the instance or bucket. Then, configure the **Sensitive data sampling** and **Audit log archiving** parameters.

Parameter	Description
Identify permissions	The permissions to detect sensitive data in selected data assets.
Desensitization permissions	The permissions to mask sensitive data in selected data assets.
Audit permissions	The permissions to audit data in selected data assets.
Sensitive data sampling	The number of sensitive data samples to reserve after SDDP detects sensitive data.If you turn on the switch in **Identify permissions**, you must also specify this parameter. If SDDP detects sensitive data in a data asset, it collects samples of the detected data. You can use the sensitive data samples for further analysis. Valid values: **0**, **5**, and **10**.
Audit log archiving	The duration to retain the audit logs of selected data assets. If you turn on the switch in **Audit permissions**, you must also specify this parameter. Valid values: **30 days**, **90 days**, and **180 days**.

   - If you want to grant permissions on multiple instances or buckets at the same time, perform the following steps:
     a. Select the required instances or buckets and click **Batch operation**.
     b. In the **Batch processing for selected assets** dialog box, turn on or off the switches of detection, audit, and masking permissions, and configure the parameters that remain.
     c. Click **Confirm**.

   After SDDP is authorized, SDDP scans the instances to detect sensitive data.

Authorize SDDP to access self-managed databases hosted on ECS instances

A self-managed database hosted on an ECS instance must meet the following requirements before it can be scanned by SDDP:

- The ECS instance resides in a virtual private cloud (VPC).
- The database is a MySQL or SQL Server database.

1. Log on to Apsara Stack Security Center.

2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data protection authorization > Data Asset authorization**.
3. On the **Cloud hosting** page, click the **ECS self-built database** tab.
4. On the **ECS self-built database** tab, click **Add data assets**.
5. In the **Asset authorization** dialog box, configure the parameters and click **Next**.

   The following table describes the parameters.

Parameter	Description
Region	The region of the self-managed database that you want to authorize SDDP to access.
ECS instance ID	The ID of the ECS instance on which the self-managed database is hosted.
Database type	The type of the self-managed database that you want to authorize SDDP to access. Valid values: MySQL and SQL Server.
Library name	The name of the self-managed database that you want to authorize SDDP to access. **Note** If you want to authorize SDDP to access other self-managed databases hosted on the same ECS instance, click **Add Database**.
Port	The port number used to connect to the self-managed database.
User name	The username of the account that you use to connect to the self-managed database.
Password	The password of the account that you use to connect to the self-managed database.

6. In the Batch processing for selected assets dialog box, turn on or off the switches of detection, audit, and masking permissions, and configure the parameters that remain.

Parameter	Description
Identify permissions	The permissions to detect sensitive data in selected data assets.
Audit permissions	The permissions to audit data in selected data assets. SDDP allows you to collect audit logs that cover the data generation, update, and use of your data assets. The log information includes the audit rule that is triggered for a data asset, the type of the data asset, the type of the operation that triggers the audit rule, and the operator account. SDDP安全审计功能的使用，请参见Create an audit rule。
Desensitization permissions	The permissions to mask sensitive data in selected data assets.

Parameter	Description
Sensitive data sampling	The number of sensitive data samples to reserve after SDDP detects sensitive data. If you turn on the switch in **Identify permissions**, you must also specify this parameter. If SDDP detects sensitive data in a data asset, it collects samples of the detected data. You can use the sensitive data samples for further analysis. Valid values: **0**, **5**, and **10**.
Audit log archiving	The duration to retain the audit logs of selected data assets. If you turn on the switch in **Audit permissions**, you must also specify this parameter. Valid values: **30 days**, **90 days**, and **180 days**.

7. Click **Confirm**.
   After SDDP is authorized, SDDP scans the databases to detect sensitive data.

Authorize SDDP to access MaxCompute projects

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data protection authorization > Data Asset authorization**.
3. On the **Cloud hosting** page, click the **MaxCompute** tab.
4. On the **MaxCompute** tab, grant the required permissions on instances or buckets.
   - If you want to grant permissions on a single instance or bucket, turn on or off the switches in the **Identify permissions**, **Desensitization permissions**, and **Audit permissions** columns of the instance or bucket. Then, configure the **Sensitive data sampling** and **Audit log archiving** parameters.

Parameter	Description
Identify permissions	The permissions to detect sensitive data in selected data assets.
Desensitization permissions	The permissions to mask sensitive data in selected data assets.
Audit permissions	The permissions to audit data in selected data assets.
Sensitive data sampling	The number of sensitive data samples to reserve after SDDP detects sensitive data. If you turn on the switch in **Identify permissions**, you must also specify this parameter. If SDDP detects sensitive data in a data asset, it collects samples of the detected data. You can use the sensitive data samples for further analysis. Valid values: **0**, **5**, and **10**.
Audit log archiving	The duration to retain the audit logs of selected data assets. If you turn on the switch in **Audit permissions**, you must also specify this parameter. Valid values: **30 days**, **90 days**, and **180 days**.

   - If you want to grant permissions on multiple instances or buckets at the same time, perform the following steps:
     a. Select the required instances or buckets and click **Batch operation**.
     b. In the **Batch processing for selected assets** dialog box, turn on or off the switches of detection, audit, and masking permissions, and configure the parameters that remain.

5. Run the following commands on the MaxCompute client to add the SDDP account **yundun_sddp** to the MaxCompute project. SDDP uses this account to access the MaxCompute project.

```
add user aliyun$yundun_sddp;
grant admin to aliyun$yundun_sddp;
```

You can perform one of the following operations based on the returned result:

- If no error messages are returned, go to .
- If an error message is returned, go to .

6. (Optional)Run the following command on the MaxCompute client to add the service IP addresses of SDDP to the IP address whitelist of the MaxCompute project:

If the IP address whitelist feature is enabled for the MaxCompute project, you must add the service IP addresses of SDDP to the IP address whitelist of the MaxCompute project. You can run the `setproject;` command to check whether the IP address whitelist feature is enabled. If the value of the `odps.security.vpc.whitelist=` parameter is empty, the IP address whitelist feature is not enabled. In this case, you can skip this step.

```
setproject odps.security.ip.whitelist=11.193.236.0/24,11.193.64.0/24,11.193.58.0/24 odps.security.vpc.whitelist=<VPC ID>;
// In this command, 11.193.236.0/24, 11.193.64.0/24, and 11.193.58.0/24 are the Classless Inter-Domain Routing (CIDR) blocks used by SDDP on the classic network. You must add them to the IP address whitelist.
// Replace the VPC ID with the ID of the region where the MaxCompute project resides. The following table describes the relationships between the VPC IDs and region IDs.
```

Region	Region ID	VPC ID
China (Zhangjiakou)	cn-zhangjiakou	cn-zhangjiakou_399229
China (Beijing)	cn-beijing	cn-beijing_691047
China (Shenzhen)	cn-shenzhen	cn-shenzhen_515895
China (Shanghai)	cn-shanghai	cn-shanghai_28803
China (Hangzhou)	cn-hangzhou	cn-hangzhou_551733

> **Note** After you configure the IP address whitelist, wait for 5 minutes before you go to the next step.

7. Click **Confirm**.

> **Note** If the authorization fails, check whether the permission parameters are correctly configured and whether the SDDP account is added to the project.

After SDDP is authorized, SDDP scans the projects to detect sensitive data.

In the list of projects on which SDDP has access permissions, you can revoke permissions on the projects. If you revoke permissions on a project, SDDP no longer scans the project.

Authorize SDDP to access AnalyticDB for MySQL clusters

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data protection authorization > Data Asset authorization**.

3. On the **Cloud hosting** page, click the **ADS** tab.
4. On the **ADS** tab, click **Add data assets**.
5. In the **Add data assets** dialog box, configure the parameters and click **Confirm**.

   The following table describes the parameters.

Parameter	Description
Region	The region of the AnalyticDB for MySQL database that you want to authorize SDDP to access.
Instance Name	The name of the cluster to which the AnalyticDB for MySQL database belongs.
Database Name	The name of the AnalyticDB for MySQL database.
User name	The username and password of the account that you use to connect to the AnalyticDB for MySQL database.
Password	
Automatic scanning	The switch of triggering scans on the AnalyticDB for MySQL database each time identification rule settings are modified.

6. On the **ADS** tab, grant the required permissions on multiple instances or buckets at the same time.
   - If you want to grant permissions on a single instance or bucket, turn on or off the switches in the **Identify permissions**, **Desensitization permissions**, and **Audit permissions** columns of the instance or bucket. Then, configure the **Sensitive data sampling** and **Audit log archiving** parameters.

Parameter	Description
Identify permissions	The permissions to detect sensitive data in selected data assets.
Desensitization permissions	The permissions to mask sensitive data in selected data assets.
Audit permissions	The permissions to audit data in selected data assets.
Sensitive data sampling	The number of sensitive data samples to reserve after SDDP detects sensitive data.If you turn on the switch in **Identify permissions**, you must also specify this parameter. If SDDP detects sensitive data in a data asset, it collects samples of the detected data. You can use the sensitive data samples for further analysis. Valid values: **0**, **5**, and **10**.
Audit log archiving	The duration to retain the audit logs of selected data assets. If you turn on the switch in **Audit permissions**, you must also specify this parameter. Valid values: **30 days**, **90 days**, and **180 days**.

   - If you want to grant permissions on multiple instances or buckets at the same time, perform the following steps:
     a. Select the required instances or buckets and click **Batch operation**.
     b. In the **Batch processing for selected assets** dialog box, turn on or off the switches of detection, audit, and masking permissions, and configure the parameters that remain.
     c. Click **Confirm**.

After SDDP is authorized, SDDP scans the databases to detect sensitive data.

Authorize SDDP to access AnalyticDB for PostgreSQL instances

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data protection authorization > Data Asset authorization**.
3. On the **Cloud hosting** page, click the **GPDB** tab.
4. On the **GPDB** tab, click **Add data assets**.
5. In the **Add data assets** dialog box, configure the parameters and click **Confirm**.

   The following table describes the parameters.

Parameter	Description
Region	The region of the AnalyticDB for PostgreSQL database that you want to authorize SDDP to access.
Instance Name	The name of the instance that hosts the AnalyticDB for PostgreSQL database.
Database Name	The name of the AnalyticDB for PostgreSQL database.
User name	The username and password of the account that you use to connect to the AnalyticDB for PostgreSQL database.
Password	
Automatic scanning	The switch of triggering scans on the AnalyticDB for PostgreSQL database each time identification rule settings are modified.

6. On the **GPDB** tab, grant the required permissions on multiple instances or buckets at the same time.
   - If you want to grant permissions on a single instance or bucket, turn on or off the switches in the **Identify permissions**, **Desensitization permissions**, and **Audit permissions** columns of the instance or bucket. Then, configure the **Sensitive data sampling** and **Audit log archiving** parameters.

Parameter	Description
Identify permissions	The permissions to detect sensitive data in selected data assets.
Desensitization permissions	The permissions to mask sensitive data in selected data assets.
Audit permissions	The permissions to audit data in selected data assets.
Sensitive data sampling	The number of sensitive data samples to reserve after SDDP detects sensitive data.If you turn on the switch in **Identify permissions**, you must also specify this parameter. If SDDP detects sensitive data in a data asset, it collects samples of the detected data. You can use the sensitive data samples for further analysis. Valid values: **0**, **5**, and **10**.
Audit log archiving	The duration to retain the audit logs of selected data assets. If you turn on the switch in **Audit permissions**, you must also specify this parameter. Valid values: **30 days**, **90 days**, and **180 days**.

- If you want to grant permissions on multiple instances or buckets at the same time, perform the following steps:
  a. Select the required instances or buckets and click **Batch operation**.
  b. In the **Batch processing for selected assets** dialog box, turn on or off the switches of detection, audit, and masking permissions, and configure the parameters that remain.
  c. Click **Confirm**.

After SDDP is authorized, SDDP scans the databases to detect sensitive data.

Authorize SDDP to access ApsaraDB for OceanBase clusters

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data protection authorization > Data Asset authorization**.
3. On the **Cloud hosting** page, click the **OceanBase** tab.
4. On the **OceanBase** tab, click **Add data assets**.
5. In the **Add data assets** dialog box, configure the parameters and click **Next**.

   The following table describes the parameters.

Parameter	Description
Region	The region of the ApsaraDB for OceanBase database that you want to authorize SDDP to access.
Database type	The type of the ApsaraDB for OceanBase database. Valid values: MySQL and Oracle.
Cluster Name	The name of the cluster to which the ApsaraDB for OceanBase database belongs.
Tenant Name	The name of the tenant to which the ApsaraDB for OceanBase database belongs.
Database Name	The name of the ApsaraDB for OceanBase database. **Note** If you want to authorize SDDP to access other ApsaraDB for OceanBase databases hosted on the same ECS instance, click **Add Database**.
Link Address	The endpoint that you use to connect to the ApsaraDB for OceanBase database.
User name	The username and password of the account that you use to connect to the ApsaraDB for OceanBase database.
Password	

6. In the Batch processing for selected assets dialog box, turn on or off the switches of detection, audit, and masking permissions, and configure the parameters that remain.

Parameter	Description
Identify permissions	The permissions to detect sensitive data in selected data assets.

Parameter	Description
Audit permissions	The permissions to audit data in selected data assets. SDDP allows you to collect audit logs that cover the data generation, update, and use of your data assets. The log information includes the audit rule that is triggered for a data asset, the type of the data asset, the type of the operation that triggers the audit rule, and the operator account. SDDP安全审计功能的使用，请参见Create an audit rule。
Desensitization permissions	The permissions to mask sensitive data in selected data assets.
Sensitive data sampling	The number of sensitive data samples to reserve after SDDP detects sensitive data. If you turn on the switch in **Identify permissions**, you must also specify this parameter. If SDDP detects sensitive data in a data asset, it collects samples of the detected data. You can use the sensitive data samples for further analysis. Valid values: **0**, **5**, and **10**.
Audit log archiving	The duration to retain the audit logs of selected data assets. If you turn on the switch in **Audit permissions**, you must also specify this parameter. Valid values: **30 days**, **90 days**, and **180 days**.

7. Click **Confirm**.
   After SDDP is authorized, SDDP scans the databases to detect sensitive data.

Import the logon information for multiple data assets at the same time

SDDP allows you to upload an Excel file to import the logon information for multiple data assets, including RDS databases, PolarDB-X databases, and self-managed databases hosted on ECS instances, at the same time to improve authorization efficiency. The following procedure describes how to import the logon information for multiple data assets at the same time:

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data protection authorization > Data Asset authorization**.
3. On the **Cloud hosting** page, click **Batch password import** in the upper-right corner.
4. In the **Batch password import** dialog box, click **SDDP Authorization File Template.xlsx**.
5. Open the downloaded file, enter the usernames and passwords used to access each data asset in the **username** and **password** columns, and then save the file.

   If you modify the existing usernames and passwords in the downloaded file and upload the file to SDDP, the logon information saved in SDDP is updated.
6. In the **Batch password import** dialog box, click **File Upload** to upload the template file that you have edited.
7. Click **OK**.
   After you upload the Excel file, the usernames and passwords that you enter in the file are synchronized to the **Username** and **Password** columns for the relevant databases on the **RDS**, **DRDS**, and **ECS self-built database** tabs. Then, you can authorize SDDP to access these data assets without the need to manually enter the logon information on the **Cloud hosting** page.

27.1.10.3.3.2. Manage usernames and passwords of databases

Sensitive Data Discovery and Protection (SDDP) can detect sensitive data that is stored in a data source only after it is authorized. To authorize SDDP, you must add the username and password that are used to connect to a database of the data source. This topic describes how to view and add the username and password that are used to connect to a database.

Context

SDDP allows you to manage the usernames and passwords that are used to connect to databases of ApsaraDB RDS, PolarDB-X, and ApsaraDB for OceanBase instances.

View the username and password of a database

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data protection authorization > Authorized account management**.
3. Click the required data source. In this example, click the **RDS** tab.
4. (Optional)Specify filter conditions to search for the specific ApsaraDB RDS instance.

   > **Note** If you want to view all ApsaraDB RDS instances, skip this step.

Filter condition	Description
Region	The region where the ApsaraDB RDS instance resides.
Instance/Bucket	The name of the ApsaraDB RDS instance.
Database type	The type of the database engine that is run by the ApsaraDB RDS instance. Different data sources support different database engines: ○ ApsaraDB RDS and PolarDB-X support the MySQL and SQL Server database engines. ○ ApsaraDB for OceanBase supports the MySQL and Oracle database engines.

5. In the instance list, view the usernames and databases that are used to connect to the databases of the ApsaraDB RDS instance.

Add the username and password of a database

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data protection authorization > Authorized account management**.
3. Click the required data source. In this example, click the **RDS** tab.
4. Find the required ApsaraDB RDS instance and enter a database username and password for the **Username** and **Password** parameters.

   > **Notice** If the username or password is invalid, SDDP fails to be authorized. Make sure that the information you enter is valid.

5. Click **Add**.
   After the username and password of the database are added, **Status** of the ApsaraDB RDS instance changes to **Added successfully**.

What's next

If you want to modify the username or password of a database, find the required instance and click **Edit** in the **Actions** column.

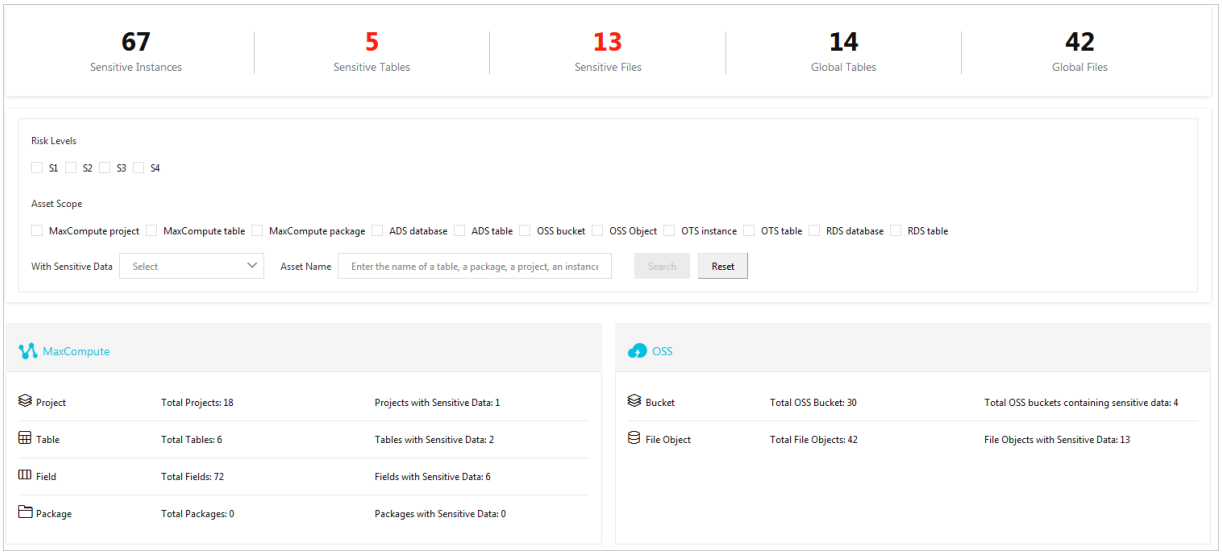
Click **Save**.

27.1.10.3.4. Sensitive data discovery

27.1.10.3.4.1. Sensitive data overview

This topic describes the Sensitive Data Overview page that displays the overall security status of your data assets.

Choose **Data Security > Sensitive Data Protection > Sensitive data Identification > Sensitive Data Overview**. On the Sensitive Data Overview page, you can view the overall security status of your data assets.



- You can view the overall information about sensitive data. The information includes the total numbers of **sensitive instances**, **sensitive tables**, **sensitive files**, **global tables**, and **global files**.
- You can search for sensitive data based on conditions such as the risk level, asset type, sensitive data type, and asset name.
- You can view the statistics on the access information and sensitive data in cloud assets, such as **MaxCompute**, **Object Storage Service (OSS)**, **AnalyticDB for MySQL**, and **Tablestore** in real time.

27.1.10.3.4.2. View statistics on sensitive data

Sensitive Data Discovery and Protection (SDDP) can detect sensitive data in data sources, such as Object Storage Service (OSS), ApsaraDB RDS, and MaxCompute. This topic describes how to view statistics on sensitive data that is detected by using SDDP.

View statistics on sensitive data detected in OSS

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection**. In the left-side navigation tree, choose **Sensitive Data Identification > Sensitive data assets**.
3. On the **OSS** tab, find the required OSS bucket and click **File Details** in the Actions column.
4. In the **OSS object query** panel, you can view the proportions of sensitive objects at each sensitivity level, top five sensitive data detection rules that are hit, and the list of objects in which the sensitive data is detected.
   - **Proportions of sensitive objects**

     In the **Proportions of sensitive objects** section, you can view the numbers of objects at the sensitivity levels of L3 (highly sensitive), L2 (moderately sensitive), L1 (low sensitive), and N/A (non-sensitive). You can also view a pie chart that shows the proportions of objects at each level.

- **Top five rules that are hit**

  In the **Top five rules that are hit** section, you can view the top five sensitive data detection rules that are hit and the number of times that each rule is hit.

- **List of objects in which the sensitive data is detected**

  In the object list, you can view the information about the objects in which the sensitive data is detected. The information includes the object name, size, type, and number of sensitive fields that are detected in the object. You can click **Hit details** in the Actions column for an object to view the details about the sensitive data detection rules that the object hits.

View statistics on sensitive data detected in ApsaraDB RDS

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection**. In the left-side navigation tree, choose **Sensitive Data Identification** > **Sensitive data assets**.
3. On the **Sensitive data assets** page, click the **RDS** tab.
4. On the **RDS** tab, find the ApsaraDB RDS instance whose details you want to view and click **Table details** in the Actions column.
5. In the **Table Query** panel, view the proportions of sensitive tables, top five sensitive data detection rules that are hit, and the list of tables in which the sensitive data is detected.
   - **Proportions of tables**

     In the **Proportions of tables** section, you can view the numbers of objects at the sensitivity levels of L3 (highly sensitive), L2 (moderately sensitive), L1 (low sensitive), and N/A (non-sensitive). You can also view a pie chart that shows the proportions of objects at each level.

   - **Top five rules that are hit**

     In the **Top five rules that are hit** section, you can view the top five sensitive data detection rules that are hit and the number of times that each rule is hit.

   - **List of tables in which the sensitive data is detected**

     In the table list, you can view the information about the tables in which the sensitive data is detected, such as the table name, total number of rows and fields, number of sensitive fields, and rules that are hit. You can click **Column details** in the Actions column to view the field details, such as the fields that hit sensitive data detection rules and the risk levels of the sensitive fields.

View statistics on sensitive data detected in MaxCompute

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection**. In the left-side navigation tree, choose **Sensitive Data Identification** > **Sensitive data assets**.
3. On the **Sensitive data assets** page, click the **MaxCompute** tab.
4. On the **MaxCompute** tab, find the required MaxCompute project and click **Table Details** in the Actions column.
5. In the **Table Query** panel, view the proportions of sensitive tables, top five sensitive data detection rules that are hit, and the list of tables in which the sensitive data is detected.
   - **Proportions of tables**

     In the **Proportions of tables** section, you can view the numbers of objects at the sensitivity levels of L3 (highly sensitive), L2 (moderately sensitive), L1 (low sensitive), and N/A (non-sensitive). You can also view a pie chart that shows the proportions of objects at each level.

   - **Top five rules that are hit**

     In the **Top five rules that are hit** section, you can view the top five sensitive data detection rules that are hit and the number of times that each rule is hit.

- **List of tables in which the sensitive data is detected**

  In the table list, you can view the information about the tables in which the sensitive data is detected, such as the table name, total number of rows and fields, number of sensitive fields, and rules that are hit. You can click **Column details** in the Actions column to view the field details, such as the fields that hit sensitive data detection rules and the risk levels of the sensitive fields.

View statistics on sensitive data detected in self-managed databases hosted on ECS instances

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection**. In the left-side navigation tree, choose **Sensitive Data Identification** > **Sensitive data assets**.
3. On the **Sensitive data assets** page, click the **ECS self-built database** tab.
4. On the **ECS self-built database** tab, find the required database and click **Table Details** in the Actions column.
5. In the **Table Query** panel, view the proportions of sensitive tables, top five sensitive data detection rules that are hit, and the list of tables in which the sensitive data is detected.
   - **Proportions of tables**

     In the **Proportions of tables** section, you can view the numbers of objects at the sensitivity levels of L3 (highly sensitive), L2 (moderately sensitive), L1 (low sensitive), and N/A (non-sensitive). You can also view a pie chart that shows the proportions of objects at each level.
   - **Top five rules that are hit**

     In the **Top five rules that are hit** section, you can view the top five sensitive data detection rules that are hit and the number of times that each rule is hit.
   - **List of tables in which the sensitive data is detected**

     In the table list, you can view the information about the tables in which the sensitive data is detected, such as the table name, total number of rows and fields, number of sensitive fields, and rules that are hit. You can click **Column details** in the Actions column to view the field details, such as the fields that hit sensitive data detection rules and the risk levels of the sensitive fields.

View statistics on sensitive data detected in PolarDB-X

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection**. In the left-side navigation tree, choose **Sensitive Data Identification** > **Sensitive data assets**.
3. On the **Sensitive data assets** page, click the **DRDS** tab.
4. On the **DRDS** tab, find the required database and click **Table Details** in the Actions column.
5. In the **Table Query** panel, view the proportions of sensitive tables, top five sensitive data detection rules that are hit, and the list of tables in which the sensitive data is detected.
   - **Proportions of tables**

     In the **Proportions of tables** section, you can view the numbers of objects at the sensitivity levels of L3 (highly sensitive), L2 (moderately sensitive), L1 (low sensitive), and N/A (non-sensitive). You can also view a pie chart that shows the proportions of objects at each level.
   - **Top five rules that are hit**

     In the **Top five rules that are hit** section, you can view the top five sensitive data detection rules that are hit and the number of times that each rule is hit.
   - **List of tables in which the sensitive data is detected**

In the table list, you can view the information about the tables in which the sensitive data is detected, such as the table name, total number of rows and fields, number of sensitive fields, and rules that are hit. You can click **Column details** in the Actions column to view the field details, such as the fields that hit sensitive data detection rules and the risk levels of the sensitive fields.

View statistics on sensitive data detected in Tablestore

You can view statistics on sensitive data detected in Tablestore.

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection**. In the left-side navigation tree, choose **Sensitive Data Identification** > **Sensitive data assets**.
3. On the **Sensitive data assets** page, click the **OTS** tab.
4. On the **OTS** tab, find the required Tablestore instance and click **Table Details** in the Actions column.
5. In the **Table Query** panel, view the proportions of sensitive tables, top five sensitive data detection rules that are hit, and the list of tables in which the sensitive data is detected.
   - **Proportions of tables**

     In the **Proportions of tables** section, you can view the numbers of objects at the sensitivity levels of L3 (highly sensitive), L2 (moderately sensitive), L1 (low sensitive), and N/A (non-sensitive). You can also view a pie chart that shows the proportions of objects at each level.
   - **Top five rules that are hit**

     In the **Top five rules that are hit** section, you can view the top five sensitive data detection rules that are hit and the number of times that each rule is hit.
   - **List of tables in which the sensitive data is detected**

     In the table list, you can view the information about the tables in which the sensitive data is detected, such as the table name, total number of rows and fields, number of sensitive fields, and rules that are hit. You can click **Column details** in the Actions column to view the field details, such as the fields that hit sensitive data detection rules and the risk levels of the sensitive fields.

View statistics on sensitive data detected in AnalyticDB for PostgreSQL

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection**. In the left-side navigation tree, choose **Sensitive Data Identification** > **Sensitive data assets**.
3. On the **Sensitive data assets** page, click the **GPDB** tab.
4. On the **GPDB** tab, find the required AnalyticDB for PostgreSQL instance and click **Table Details** in the Actions column.
5. In the **Table Query** panel, view the proportions of sensitive tables, top five sensitive data detection rules that are hit, and the list of tables in which the sensitive data is detected.
   - **Proportions of tables**

     In the **Proportions of tables** section, you can view the numbers of objects at the sensitivity levels of L3 (highly sensitive), L2 (moderately sensitive), L1 (low sensitive), and N/A (non-sensitive). You can also view a pie chart that shows the proportions of objects at each level.
   - **Top five rules that are hit**

     In the **Top five rules that are hit** section, you can view the top five sensitive data detection rules that are hit and the number of times that each rule is hit.
   - **List of tables in which the sensitive data is detected**

In the table list, you can view the information about the tables in which the sensitive data is detected, such as the table name, total number of rows and fields, number of sensitive fields, and rules that are hit. You can click **Column details** in the Actions column to view the field details, such as the fields that hit sensitive data detection rules and the risk levels of the sensitive fields.

View statistics on sensitive data detected in AnalyticDB for MySQL

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection**. In the left-side navigation tree, choose **Sensitive Data Identification** > **Sensitive data assets**.
3. On the **Sensitive data assets** page, click the **ADS** tab.
4. On the **ADS** tab, find the required AnalyticDB for MySQL cluster and click **Table Details** in the Actions column.
5. In the **Table Query** panel, view the proportions of sensitive tables, top five sensitive data detection rules that are hit, and the list of tables in which the sensitive data is detected.
   - **Proportions of tables**

     In the **Proportions of tables** section, you can view the numbers of objects at the sensitivity levels of L3 (highly sensitive), L2 (moderately sensitive), L1 (low sensitive), and N/A (non-sensitive). You can also view a pie chart that shows the proportions of objects at each level.
   - **Top five rules that are hit**

     In the **Top five rules that are hit** section, you can view the top five sensitive data detection rules that are hit and the number of times that each rule is hit.
   - **List of tables in which the sensitive data is detected**

     In the table list, you can view the information about the tables in which the sensitive data is detected, such as the table name, total number of rows and fields, number of sensitive fields, and rules that are hit. You can click **Column details** in the Actions column to view the field details, such as the fields that hit sensitive data detection rules and the risk levels of the sensitive fields.

View statistics on sensitive data detected in DataWorks

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection**. In the left-side navigation tree, choose **Sensitive Data Identification** > **Sensitive data assets**.
3. On the **Sensitive data assets** page, click the **DataWorks** tab.
4. On the **DataWorks** tab, find the required DataWorks workspace and click **Table Details** in the Actions column.
5. In the **Table Query** panel, view the proportions of sensitive tables, top five sensitive data detection rules that are hit, and the list of tables in which the sensitive data is detected.
   - **Proportions of tables**

     In the **Proportions of tables** section, you can view the numbers of objects at the sensitivity levels of L3 (highly sensitive), L2 (moderately sensitive), L1 (low sensitive), and N/A (non-sensitive). You can also view a pie chart that shows the proportions of objects at each level.
   - **Top five rules that are hit**

     In the **Top five rules that are hit** section, you can view the top five sensitive data detection rules that are hit and the number of times that each rule is hit.
   - **List of tables in which the sensitive data is detected**

In the table list, you can view the information about the tables in which the sensitive data is detected, such as the table name, total number of rows and fields, number of sensitive fields, and rules that are hit. You can click **Column details** in the Actions column to view the field details, such as the fields that hit sensitive data detection rules and the risk levels of the sensitive fields.

27.1.10.3.4.3. Query sensitive data

The Sensitive data search page displays all the sensitive data that is detected in your data assets. You can specify one or more types of sensitive data to query and view the distribution of the sensitive data across your data assets. This topic describes how to query sensitive data.

Procedure

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection**. In the left-side navigation tree, choose **Sensitive Data Identification** > **Sensitive data search**.
3. On the **Sensitive data search** page, specify filter conditions based on your business requirements.

   Sensitive Data Discovery and Protection (SDDP) provides the following filter conditions:

   - **Hit data**: the type of sensitive data. You can select multiple types at a time, such as email addresses and mobile phone numbers.
   - **Enter file name to search/Enter table name to search**: the name of the object or table in which the sensitive data is detected. Fuzzy match is supported.
   - **Region**: the region where the data asset resides.
   - **Bucket/Instance/Project**: the name of the bucket, instance, or project in which the sensitive data is detected.

   > **Note** If you specify multiple filter conditions, SDDP displays the sensitive data that meets all the specified filter conditions.

4. Click **Search**.

   In the result list, you can view the information about the objects or tables where the sensitive data is detected. The information can be grouped or sorted by using the following methods:

   - **Group sensitive data by risk level**

     On the **OSS-file** tab, set the **Sensitivity level** parameter to S1, S2, or S3. This can be used to view sensitive data at the specified sensitivity level.

   - **Sort sensitive data based on the total number of rows or sensitive fields in ascending or descending order**

     On a specific tab such as the **RDS-table** tab, click the icon to the right of **Total number of rows** or **Sensitive column**. This can be used to sort sensitive data based on the total number of rows or sensitive fields. Then, these results can be further sorted in ascending or descending order. The data is sorted in descending order after you click the icon for the first time. Then, the next time you click the icon, these results are sorted in ascending order.

5. Find the specific sensitive data, and click **Details** in the **Operation** column or click **Column details** in the **Operation** column.

   In the **Hit query** panel for a bucket or the **Column details** panel for a table, you can view the information about all the sensitive data that is detected in the current object or table. The information includes the following content:

   - **Column name**: the name of the sensitive field detected in the table.

> **Note** This parameter is displayed only in the **Column details** panel for a table in an ApsaraDB RDS instance, MaxCompute project, self-managed database that is hosted on an Elastic Compute Service (ECS) instance, PolarDB-X database, Tablestore instance, AnalyticDB for MySQL cluster, or AnalyticDB for PostgreSQL instance. The **Hit query** panel for an OSS bucket does not display this parameter.

- **Hit Rules**: the type and name of the sensitive data detection rule that is hit.
- **Sensitivity level**: the risk level of the detected sensitive data.
- **Number of hits**: the number of times that the sensitive data detection rule is hit in the object.

  > **Note** The **Hit query** panel for an OSS bucket does not display this parameter.

- **Data Sampling**: the samples that are collected from the data asset. If you want to configure the **Sensitive data sampling** parameter, choose **Data Security > Sensitive Data Protection > Data protection authorization > Data Asset authorization**. On the **Cloud hosting** page, you can set this parameter to 0, 5, or 10. Make sure that the value you set is the same as the value of **Sensitive data sampling** that you set when you authorize SDDP to protect your data assets.

27.1.10.3.4.4. Manage scan tasks

Sensitive Data Discovery and Protection (SDDP) is authorized to automatically scan data assets for sensitive data. On the Identify Task Monitoring page, you can view the details of scan tasks that are generated by SDDP. These tasks are used to scan data assets and rerun the scan tasks.

Context

SDDP can monitor scan tasks that are generated for sensitive data in Object Storage Service (OSS), ApsaraDB RDS, MaxCompute, self-managed databases that are hosted on Elastic Compute Service (ECS) instances, PolarDB-X, Tablestore, AnalyticDB for MySQL, and AnalyticDB for PostgreSQL.

After you authorize SDDP to access specific data assets, SDDP can be used to automatically create and run scan tasks for these data assets to detect sensitive data. By default, the **automatic scan** feature is enabled for scan tasks that you create. This feature allows SDDP to run a full scan on authorized data assets and scan the data that is newly written to or modified in these data assets at an interval of 4 hours. In addition, after you create or modify a sensitive data detection rule, SDDP automatically reruns scan tasks for which the automatic scan feature is enabled.

View the details of scan tasks

On the **Identify task monitoring** page, you can view the details of each scan task, such as the related data asset, task status, and time when the task is complete. To view the details of scan tasks, perform the following steps:

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Sensitive Data Identification > Identify task monitoring**.
3. On the Identify task monitoring page, click the tab of the data asset for which you want to view scan tasks.
4. (Optional)Select the region, enter the name of the data asset, specify the start and end of the time range to query, and then click **Search**.
5. In the task list, view the details of each scan task, such as the related data asset, task status, and time when the task is complete.

Rescan your data assets

You can rerun scan tasks in the following scenarios:

- If the **automatic scan** feature is not enabled for a scan task, the scan task is not run after the task is created. In this case, you must rerun the scan task.
- If you enable the **automatic scan** feature for a scan task, SDDP can be used to automatically rerun the scan task and scan the data that is newly written to or modified in the specific data asset at an interval of 4 hours. You can also rerun the scan task to immediately scan the specific data asset after you modify the data in the data asset.

To rescan a data asset for sensitive data, perform the following steps:

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Sensitive Data Identification > Identify task monitoring**.
3. On the Identify task monitoring page, click the tab of the data asset for which you want to rescan data.
4. Find the specific data asset and click **Rescan** in the **Operation** column.
5. In the **Confirm rescan** dialog box, click **OK**.

   In most cases, the rescan process requires about 10 minutes to complete. Wait until the data asset is scanned.

   After the rescan is started, the scan task enters the **Scanning** or **Waiting** state. The percentage that appears in the **Scan Status** column indicates the progress of the scan task.

What's next

After the scan is complete, the scan task enters the **Complete** state. To view the latest scan results, choose **Data Security > Sensitive Data Protection > Sensitive Data Identification > Sensitive data assets** and click the tab of the data asset for which you want to view the scan results.

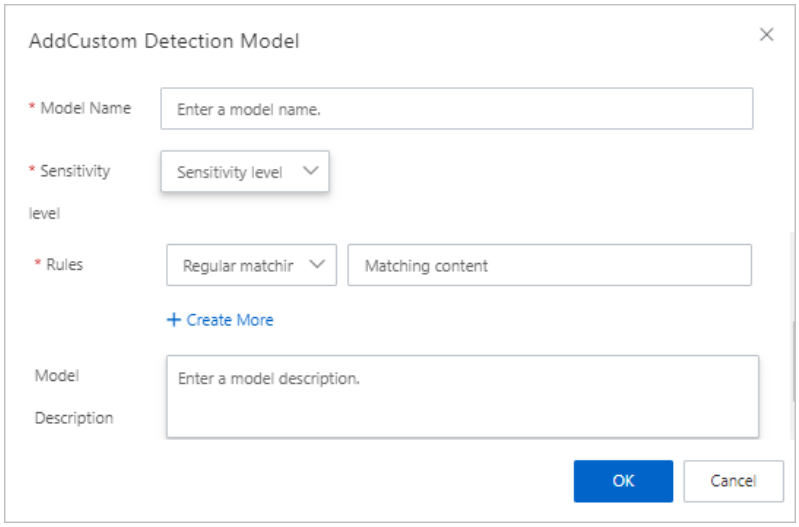
27.1.10.3.4.5. Manage detection rules

Sensitive Data Discovery and Protection (SDDP) allows you to customize the detection rules for classifying sensitive data. You can view and configure detection rules to detect sensitive data. This topic describes how to create and manage custom detection rules, view built-in detection rules, and modify sensitivity levels.

Create a custom detection rule

SDDP detects sensitive data in files or tables based on specified rules and generates alerts. You can customize detection rules to detect sensitive data based on your business requirements. To customize a detection rule, perform the following steps:

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Sensitive Data Identification > Identification Rules**.
3. On the **Identification Rules** page, click **Add Rule**.
4. In the **Add Rule** panel, configure parameters.



The following table describes the parameters to create a custom detection rule.

Parameter	Description
Rule name	The name of the detection rule.
Rule source	The source of the detection rule. The default value is **Customize** and cannot be changed.
Rule type	The type of the detection rule. Valid values: ○ **Algorithm**: Select an algorithm to detect sensitive data. ○ **Regular expression**: Select a regular expression to detect sensitive data. **Note** ○ If you have not created custom detection rules, SDDP detects sensitive data based on algorithm-based built-in detection rules. ○ The built-in detection rules that SDDP provides apply to various types of common sensitive data, including mobile phone numbers and ID card numbers. We recommend that you view built-in detection rules to check whether the sensitive data that you want to detect can be detected by built-in detection rules before you create the custom detection rule. For more information, see View built-in detection rules.
Sensitivity level	The sensitivity level for the detection rule. Valid values: ○ **N/A**: non-sensitive ○ **L1**: low sensitive ○ **L2**: moderately sensitive ○ **L3**: highly sensitive

Parameter	Description
Rule classification	The class of the sensitive data that the detection rule can detect. Valid values: ○ Personal and sensitive information ○ Sensitive device information ○ Sensitive key information ○ Sensitive picture information ○ Sensitive corporate information ○ Sensitive location information ○ Universal sensitive information
Rules	The content of the detection rule. The content is used to match sensitive fields or text. This parameter is based on the **Rule type** parameter. ○ If the **Rule type** parameter is set to **Keywords**, you must set the **Method** parameter and enter the keyword that is used to detect sensitive data in the **Keywords** field. If you want to create a custom detection rule to detect the mobile phone number *1331234*****. You can set the **Method** parameter to **Contains** and enter *1331234***** in the **Keywords** field. **Note** The keyword must be a precise value, such as a specific mobile phone number, email address, or ID card number. ○ If the **Rule Type** parameter is set to **Regular Expression**, you must enter the regular expression that is used to detect sensitive data in the **Regular Expression** field. If you want to create a custom detection rule to detect mobile phone numbers, you can enter *^((13[0-9])|(14[5,7])|(15[0-3,5-9])|(17[0,3,5-8])|(18[0-9])|166|198|199|(147))\\d{8}$* in the **Regular Expression** field. **Note** After a detection rule is created, the detection rule appears in the rule list. However, the rule list does not display the details of the rule. You can click **Details** in the Operation column to view the details of the detection rule.

5. Click **Enable** or **Save**.
   - **Enable**: If you click **Enable**, the detection rule is created and enabled. SDDP starts to detect sensitive data based on the detection rule.
   - **Save**: If you click **Save**, the detection rule is created but not enabled. To enable the detection rule, you must turn on the switch in the Status column for the detection rule in the rule list.

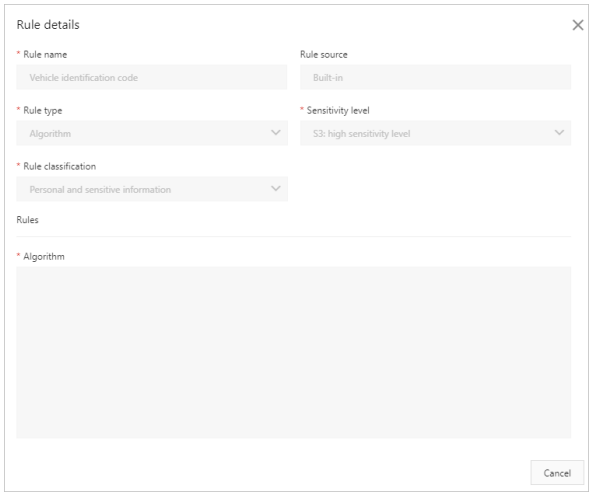
Note
- SDDP detects sensitive data based on all sensitive data detection rules that are enabled.
- A detection rule takes effect after it is created and enabled. If you want to temporarily exclude specific data from sensitive data, you can disable the specific detection rule. After you disable a detection rule, SDDP no longer detects sensitive data based on the detection rule. We recommend that you enable all detection rules to reduce risks.
- You can modify and delete custom detection rules. You can view built-in detection rules but cannot modify or delete them.

View built-in detection rules

The built-in detection rules that SDDP provides apply to various types of common sensitive data, such as mobile phone numbers and ID card numbers. You can view all information about a built-in detection rule, such as the rule type, rule name, and sensitivity level. You cannot view the rule definition. To view built-in detection rules, perform the following steps:

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Sensitive Data Identification > Identification Rules**.
3. On the **Identification Rules** page, set the **Rule Source** parameter to **Built-in**.
4. View built-in detection rules in the list that appears.

   You can view the information about each built-in detection rule, such as **Rule name**, **Rule classification**, **Rule source**, and **Rule type**.
5. To view the details about a specific built-in detection rule, find the built-in detection rule and click **Details** in the Operation column.
6. In the **Rule details** dialog box, view the details of the built-in detection rule.



   You can view **Rule name**, **Rule source**, **Rule type**, **Sensitivity level**, and **Rule classification** of a built-in detection rule. You cannot view the **algorithm** or **regular expression** of a built-in detection rule.

Modify a sensitivity level

SDDP allows you to modify the name and description of a sensitivity level. To modify a sensitivity level, perform the following steps:

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Sensitive Data Identification > Identification Rules**.
3. On the Identification Rules page, click the **Level settings** tab.
4. Find the sensitivity level that you want to modify and click **Edit** in the Actions column.
5. In the **Sensitivity level** dialog box, modify the information in the Sensitivity level and Description fields.

   By default, SDDP marks sensitive data with the following sensitivity levels: **N/A**, **S1**, **S2**, and **S3**. **N/A** indicates an unknown risk level. The sensitivity level of S1, S2, and S3 increases in sequence. You can customize the names and descriptions of the four sensitivity levels to classify the sensitive data detected in your data assets based on your business requirements. SDDP provides the following default descriptions for the S1, S2, and S3 levels:

   - **S1**: low risk.

- **S2**: medium risk.
- **S3**: high risk.

6. Click **Confirm**.
   The modification immediately takes effect after you submit it. Refresh the **Identification Rules** page. You can view the new sensitivity level on the **Level settings** tab.

27.1.10.3.5. Check data permissions

27.1.10.3.5.1. View permission statistics

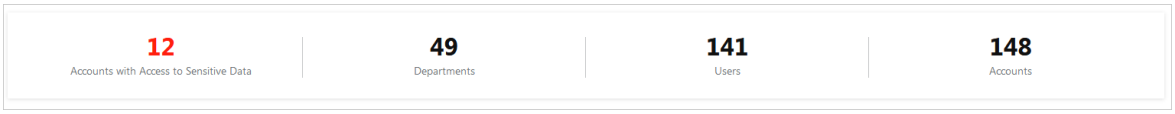
This topic describes how to view permission statistics.

Context

On the Permission Management page, you can view the overall permission distribution of Apsara Stack. You can also identify vulnerable accounts and users, and troubleshoot and handle security issues at your earliest opportunity.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Data Security > Security Data Protection > Data Permissions > Permission Management**.
3. View the overall permission statistics.



   - **Accounts with Access to Sensitive Data**: the number of accounts that can access sensitive data.
   - **Departments**: the number of departments in Apsara Stack.
   - **Users**: the number of users in Apsara Stack.
   - **Accounts**: the number of accounts in Apsara Stack.
4. View the department-level permission statistics.
   You can view the statistics on the users, accounts, and anomalous events that are related to permissions for each department.

27.1.10.3.5.2. View the permissions of an account

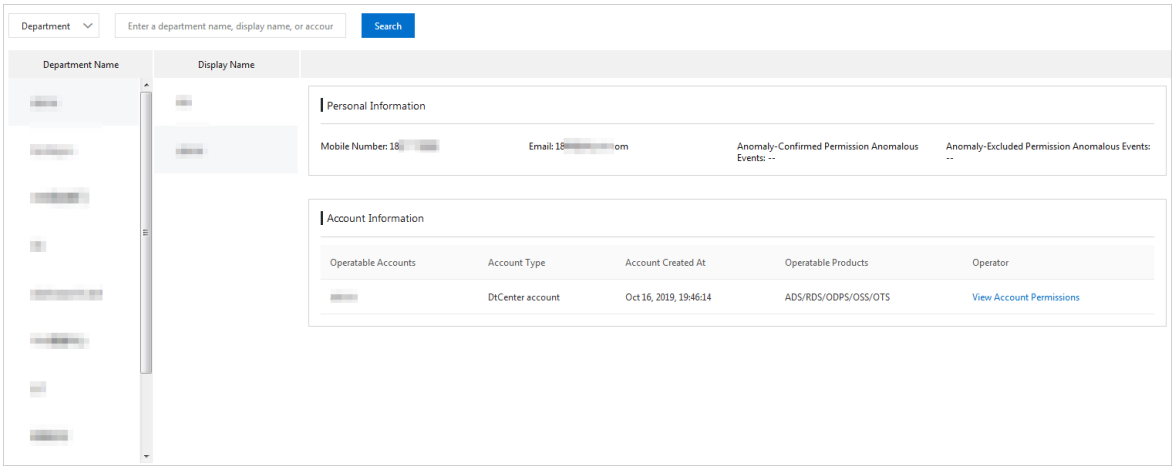
This topic describes how to view the permissions of an account.

Context

You can search for an account to view its information so that you can quickly find the owner of sensitive data.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Data Security > Security Data Protection > Data Permissions > Permission Search**.



3. Search for a specific account.

   To search for an account, perform the following steps:

   i. Select **Department** or **Employee** from the drop-down list.

   ii. Enter a keyword, such as a department name or account.

   iii. Click **Search** to view the search results in the **Display Name** column.

   > **Note** You can also click a department in the Department Name column. All accounts of the department are listed in the Display Name column.

4. In the **Display Name** column, click the specific account.
5. View information in the **Personal Information** and **Account Information** sections on the right.
   - **Personal Information**

     You can view the contact information about the account owner, and the number of confirmed anomalous events that are related to permission access. You can also view the number of excluded anomalous events that are related to permission access.
   - **Account Information**

     You can view the accounts that the owner can use, the types and the time when the accounts are created, and Apsara Stack services that the accounts can access.

     You can click **View Account Permissions** in the Actions column of an account to view the resources, resource types, resource paths, and operation permissions.

27.1.10.3.6. Monitor data flows

27.1.10.3.6.1. View data flows in DataHub

This topic describes how to view data flows in DataHub.

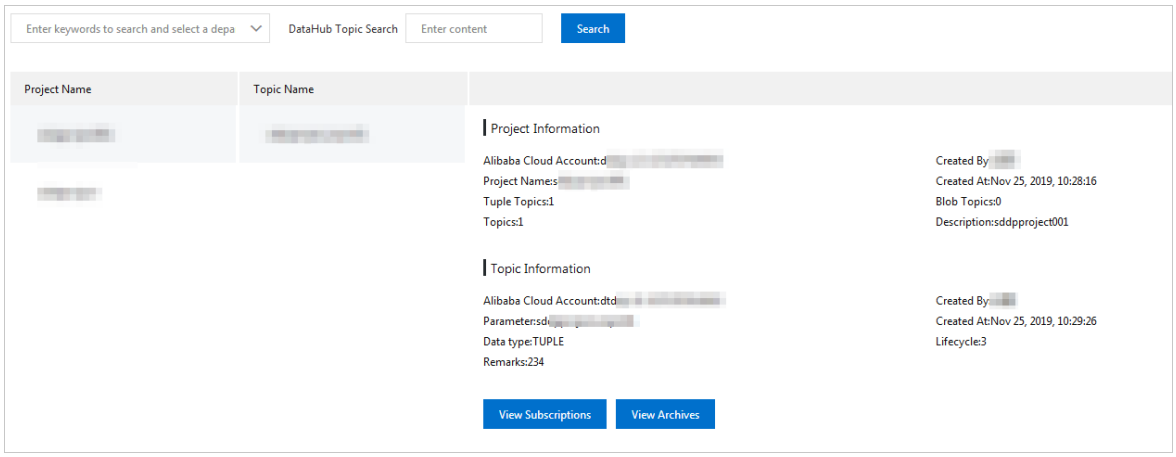
Context

DataHub is a platform that is designed to process streaming data. You can publish and subscribe to streaming data in DataHub, and distribute the data to other platforms. DataHub allows you to analyze streaming data and build applications based on the streaming data.

On the **DataHub** page, you can view the details about data flows in DataHub. The details include the relationships between DataHub projects and topics, and the relationships among topics, subscribed applications, and archive sources.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Data Security > Security Data Protection > Data Flow Monitoring > DataHub**.
3. In the search box, enter a keyword and select a department from the drop-down list. Enter a topic keyword in the **DataHub Topic Search** field and click **Search**.



> **Note**
>
> You can also click the required project in the **Project Name** column and click the required topic in the **Topic Name** column.

In the **Project Information** and **Topic Information** sections, you can view the information about the project and the topic.

- **Project Information**

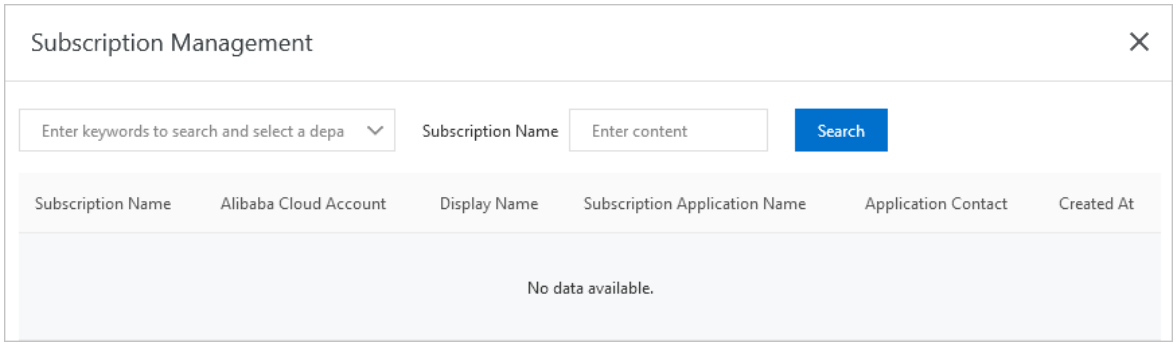
  Displays information such as the project name, Apsara Stack account, creator, creation time, and number of topics.

- **Topic Information**

  Displays information such as the topic name, Apsara Stack account, creator, creation time, and topic type.

4. Click **View Subscriptions** to view the subscription list.

   The subscription list contains information such as the subscription name, Apsara Stack account of the creator, display name, name of the subscribed application, and application contact.



   i. Enter a keyword and select a department from the drop-down list.
   ii. Enter a keyword in the **DataHub Topic Search** field.
   iii. Click **Search** to search for the required DataHub topic.

5. Click **View Archives** to view the archive list.

   The archive list contains information such as the name of the connected instance, Apsara Stack account of the creator, display name, source service, resource path, and risk level.

   i. Enter a keyword and select a department from the drop-down list.

   ii. Enter a keyword in the **DataHub Topic Search** field.

   iii. Click **Search** to find the required DataHub topic.

27.1.10.3.6.2. View data flows in Data Integration

This topic describes how to view data flows in Data Integration.

Context

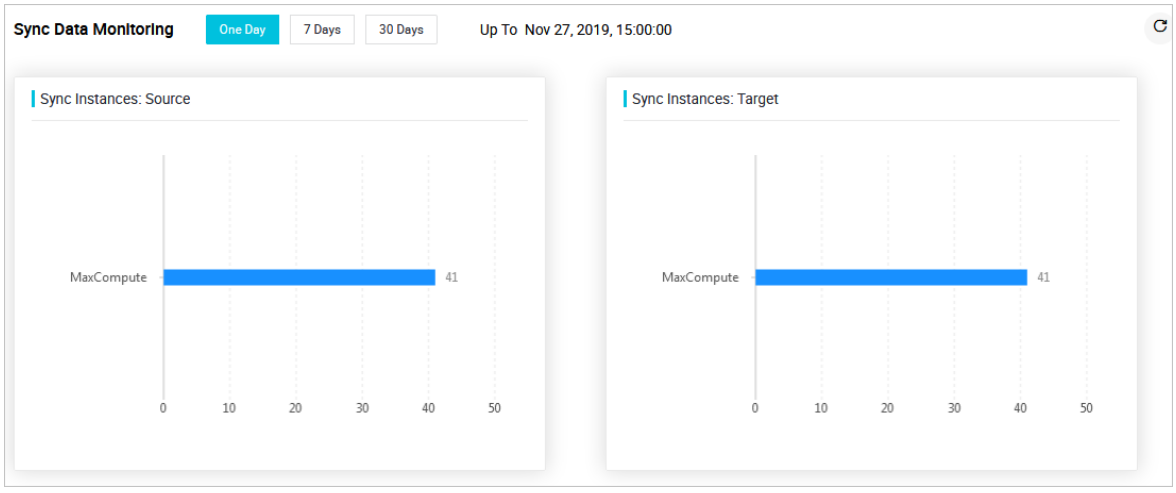
DataWorks is a comprehensive, professional cloud research and development (R&D) platform for big data. DataWorks is used as an operating system that delivers intelligent, efficient, secure, and reliable big data services. DataWorks meets your requirements for data and quality management. It can also be used to provide data services for external systems.

DataWorks includes the Data Integration feature that provides a stable, efficient, and scalable data synchronization platform on Apsara Stack. Data Integration implements fast and stable data transmission and synchronization between various data sources in complex networks.

Procedure

1.
2. In the left-side navigation pane, choose **Data Security > Security Data Protection > Data Flow Monitoring > CDP Flow Monitoring**.
3. On the **Data Integration** page, click **Sync Data Monitoring**.
4. View the statistical graph for the number of synchronized instances.

   > **Note**
   > - The number of synchronized instances is measured from two aspects: **Sync Instances: Source** and **Sync Instances: Target**.
   > - You can view statistics within different periods (**One Day**, **7 Days**, and **30 Days**).

5. In the **Sync Instances** section, view information such as the ID, time, node name, data type, and the amount of synchronized data.

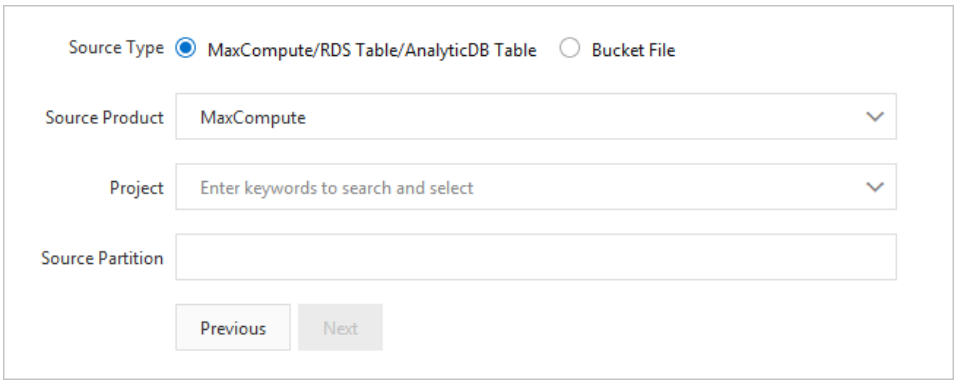
27.1.10.3.7. Sensitive data masking

27.1.10.3.7.1. Create a static masking task

This topic describes how to create a static masking task and run the task to mask sensitive data.

Procedure

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Sensitive Data Desensitization > Static Desensitization**.
3. In the upper-right corner of the **Desensitization task configuration** tab, click **Add Desensitization Task**.
4. On the **Add Desensitization Task** page, configure parameters.
   i. In the **Basic Task Information** step, specify **Task Name** and **Task notes**. Then, click **Next**.
   ii. In the **Desensitization Source Configuration** step, specify the source of data that you want to mask and click **Next**.



You can use Sensitive Data Discovery and Protection (SDDP) to create masking tasks for the following data sources: tables in ApsaraDB RDS, MaxCompute, AnalyticDB for MySQL, and ApsaraDB for OceanBase. You can also create masking tasks for objects in Object Storage Service (OSS). You cannot create masking tasks for tables in PolarDB. The following table describes the parameters used to create a masking task for each data source.

Data source	Parameter	Description
MaxCompute	**Types of data storage**	Select **RDS Table / MaxCompute Table /PolarDB Table /ADS Table /OceanBase Table**.
	Source Product	Select **MaxCompute**.
	Source Database/Project	Select the source database or project whose data you want to mask from the drop-down list.
	Source table name	Select the source table whose data you want to mask from the drop-down list.

Data source	Parameter	Description
	Source Partition	Enter the name of the source partition whose data you want to mask. You can configure partitions when you create a MaxCompute table. Partitions define different logical divisions of a table. When you query data, you can specify partitions to improve query efficiency **Note** **Source Partition** is optional. If you leave this parameter unspecified, SDDP masks sensitive data in all partitions of the source table.
ApsaraDB RDS	**Types of data storage**	Select **RDS Table / MaxCompute Table /PolarDB Table /ADS Table /OceanBase Table**.
	Source Product	Select **RDS**.
	Source Database/Project	Select the source database or project whose data you want to mask from the drop-down list.
	Source table name	Select the source table whose data you want to mask from the drop-down list.
	Sample SQL	Optional. Enter an SQL statement and specify the data that you want to mask.
ApsaraDB for OceanBase	**Types of data storage**	Select **RDS Table / MaxCompute Table /PolarDB Table /ADS Table /OceanBase Table**.
	Source Product	Select **OceanBase**.
	Source Database/Project	Select the source database or project whose data you want to mask from the drop-down list.
	Source table name	Select the source table whose data you want to mask from the drop-down list.
	Sample SQL	Optional. Enter an SQL statement and specify the data that you want to mask.
ADS	**Types of data storage**	Select **RDS Table / MaxCompute Table /PolarDB Table /ADS Table /OceanBase Table**.
	Source Product	Select **ADS**.
	Source Database/Project	Select the source project whose data you want to mask from the drop-down list.
	Source table name	Select the source table whose data you want to mask from the drop-down list.
	Types of data storage	Select **OSS files**.

Data source	Parameter	Description
OSS	**File source**	Upload a file from your computer or select a bucket. Valid values: ■ **Uploaded Local File**: If you select this option, click **Select a local file** and select a source file from your computer. ■ **OSS Bucket**: If you select this option, select the OSS bucket to which the source object belongs.
	Source file description	Enter an informative description for the bucket to help identify the task. **Note** This parameter is required only when you set **File source** to **Uploaded Local File**.
	OSS Bucket where the source file is located	Select the OSS bucket to which the source object belongs. **Note** This parameter is required only when you set **File source** to **OSS Bucket**.
	Source file names	Optional. Enter the name of the source object. **Note** This parameter is required only when you set **File source** to **OSS Bucket**. If you want to use wildcards to specify objects, turn on **Open the pass**. After you turn on Open the pass, you can use asterisks (*) as wildcards to specify multiple OSS objects at a time. However, you can use asterisks only in object names. For example, enter test*.xls. After you specify an object name by using an asterisk, SDDP masks the data of the matched objects. Make sure that these objects use the same column structure.
	Separator selection	Optional. Select a column delimiter based on the format of the object that you specify. This parameter is required for objects in the CSV or TXT format. Valid values: ■ **Semicolon ";" (MacOS/Linux default)** ■ **Comma "," (Windows default)**
	Table contains header rows	Optional. Specify whether the data to be masked contains header rows.

iii. In the **Desensitization algorithm** step, specify the algorithm to mask data and click **Next**.

In this step, you must specify the algorithm type, select an algorithm, and turn on the masking switch for the source field of data that you want to mask.

iv. (Optional)In the **Data Watermark** step, turn on **Open data watermark**. Specify the following parameters: Please select the field to embed the watermark, Please select a watermark algorithm, and Please enter watermark information. Then, click **Next**.

The Please select a watermark algorithm parameter has the following values:

- **Space Algorithm**: If you want to add watermarks for fields of a string type, select this value.
- **Modify the least significant bit algorithm**: If you want to add watermarks for fields of a numeric type, select this value.

v. In the **Destination Location Configuration** step, specify the destination table to store the data after masking, test and make sure that you have write permissions on the destination table, and then click **Next**. The parameters for the destination table include **Types of data storage** and **Target**.

vi. In the **Confirm Process Logic** step, configure the processing logic of the task.

Parameter	Description
How the task is triggered	Select a method to run the masking task. Valid values: ■ **Manual Only**: You must manually run the masking task on the Static Desensitization page. ■ **Scheduled Only**: The masking task is automatically run at a specific point in time on an hourly, daily, or monthly basis. ■ **Manual + Scheduled**: You can manually run the masking task or enable automatic running of the masking task at a specific point in time on an hourly, daily, or monthly basis.
Turn on incremental desensitization	Optional. Enable incremental masking based on your business requirements. If you turn on this switch, SDDP masks only the data that is added after the last masking task is completed. You must specify a field whose value is increased over time as the incremental identifier. For example, you can specify the creation time field or the auto-increment ID field as the incremental identifier. **Note** SDDP supports incremental masking only for data in ApsaraDB RDS.
Shard field	Optional. Select a field based on which SDDP divides the source data into multiple shards and concurrently masks the data in these shards. In this case, data masking is more efficient. You can specify one or more shard fields based on your business requirements. **Note** ■ SDDP supports incremental masking only for data in ApsaraDB RDS. We recommend that you use a primary key or a field on which a unique index is created as the shard field. ■ If you leave this parameter unspecified, a primary key is used as the shard field. SDDP divides the source data based on the primary key and masks the data. If the source data does not have a primary key, you must specify a shard field. Otherwise, the masking task fails. ■ If you specify excessive shard fields, query performance and data accuracy may deteriorate. Proceed with caution.

Parameter	Description
Table name conflict resolution	Select a method to handle a table name conflict. Valid values: ■ **Delete the target table and create a new table with the same name.** ■ **Attach data to the target table**: This method is recommended.
Row Conflict Resolution	Select a method to handle a row conflict. Valid values: ■ **Keep conflicting rows in the target table and discard the new data**: This method is recommended. ■ **Delete conflicting rows in the target table and insert the new data.**

vii. Click **Submit**.

After you create the masking task, you can view the task in the list of masking tasks on the **Desensitization task configuration** tab.

5. In the list of masking tasks, turn on the switch and run the masking task.
6. On the **Task Execution Status** tab, view **Execution Progress** and **Status** of the masking task.

27.1.10.3.7.2. View dynamic data masking tasks

Sensitive Data Discovery and Protection (SDDP) provides the dynamic data masking feature. You can call the ExecDatamask operation to dynamically mask sensitive data.

Context

When you call this operation, you must specify the ID of the data masking template to use. Static data masking and dynamic data masking can use the same template. To obtain the template ID, you must log on to Apsara Stack Security Center and choose **Data Security > Sensitive Data Protection > Sensitive Data Desensitization > Desensitization Template**. You can also create custom data masking templates. For more information, see Create a data masking template.

Limits

Before you can call the ExecDatamask operation to dynamically mask sensitive data, make sure that the size of the sensitive data is less than 2 MB. The size is specified by the `Data` parameter.

View the call history of the ExecDatamask operation

Log on to Apsara Stack Security Center. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Sensitive Data Desensitization > Dynamic desensitization**. On the page that appears, you can view the call history of the ExecDatamask operation. Each record includes the name of the operation, the UID of the Alibaba Cloud account or Resource Access Management (RAM) user that was used to call the operation, the IP address of the user who initiated the call, the times at which the operation was first and last called, and the total number of calls.

Note Only one record is generated for calls initiated by the same Alibaba Cloud account or RAM user from the same IP address. The number of calls is accumulated in this case.

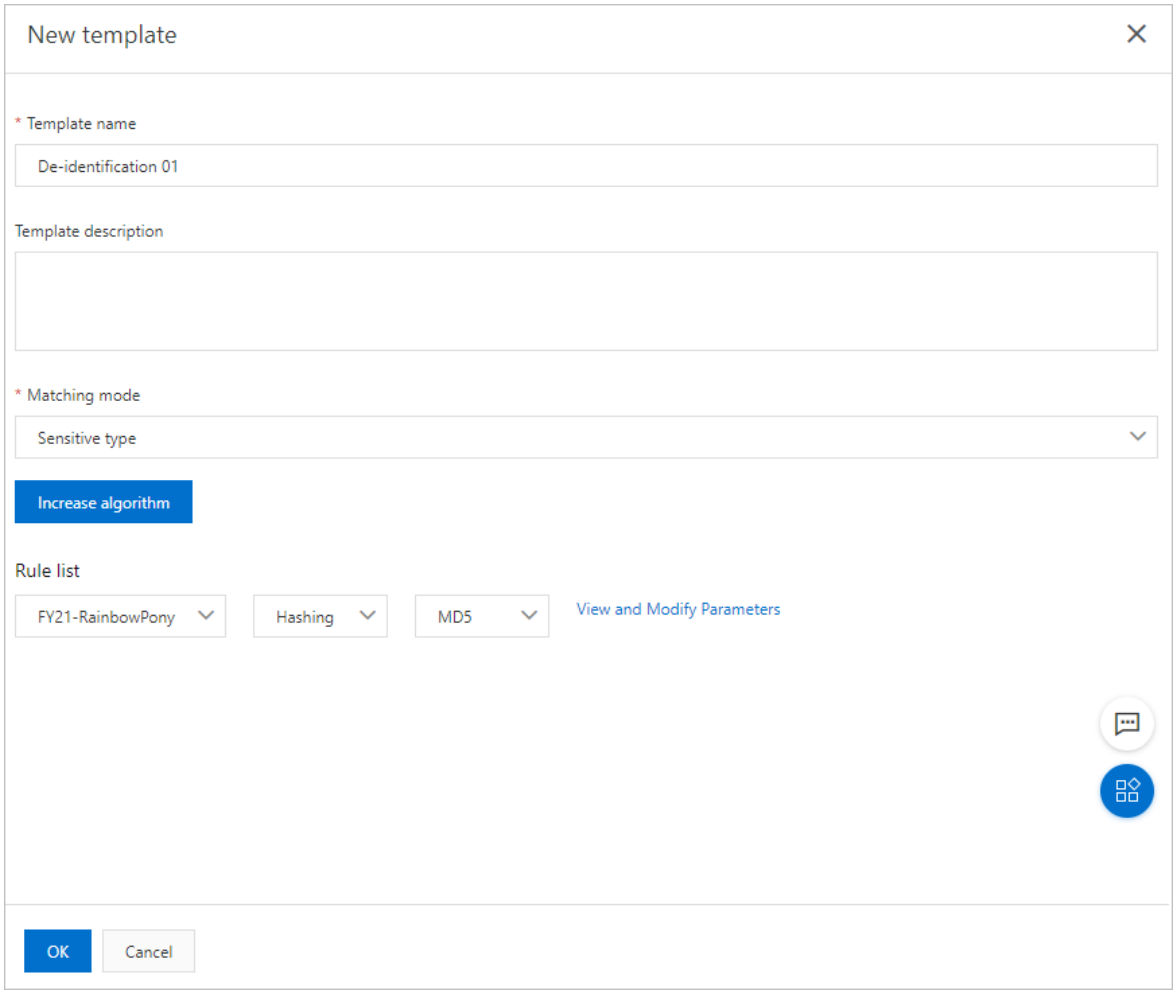
27.1.10.3.7.3. Create a data masking template

Sensitive Data Discovery and Protection (SDDP) allows you to create data masking templates. You can create a data masking template and add data masking algorithms that are frequently used in the same scenario to the template. This avoids repeated configuration of data masking algorithms and improves the efficiency in sensitive data processing. This topic describes how to create and manage data masking templates.

Create a data masking template

You can create an unlimited number of data masking templates.

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Sensitive Data Desensitization > Desensitization Template**.
3. On the **Desensitization Template** page, click **New template**.
4. In the **New template** panel, configure parameters. The following table describes the parameters to create a data masking template.



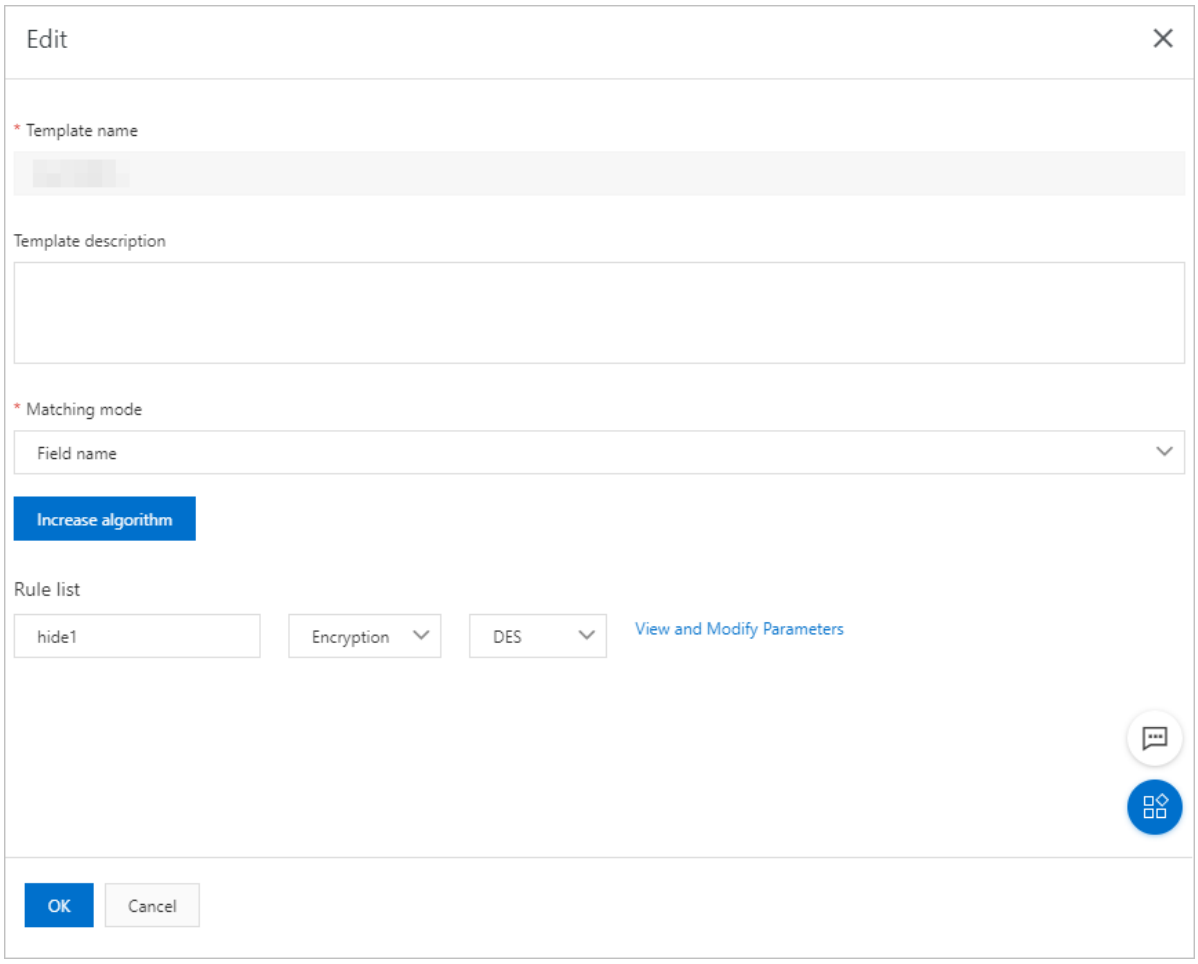
Parameter	Description
Template name	The name of the data masking template.
Template description	The description of the data masking template. You can enter information such as the scenario to which the template is applied.
Matching mode	The mode in which the data masking template handles its matched sensitive data. Valid values: ○ **Sensitive type**: If you select this option, you must select the types of sensitive data that you want to mask, such as vehicle identification numbers and unified social credit codes, and the data masking algorithm for each type of sensitive data. ○ **Field name**: If you select this option, you must specify the fields that you want to mask and the data masking algorithm for each field.

Parameter	Description
Rule list	The rules that are used to mask sensitive data. To configure a rule, select a sensitive data type or enter a field that you want to mask and specify a data masking algorithm. SDDP supports the following data masking algorithms. ○ Hashing ○ Redaction ○ Substitution ○ Rounding ○ Encryption ○ Shuffling ○ Data decryption For more information, see Configure data masking algorithms. You can configure multiple rules in a template. To configure more rules, click **Increase Algorithm**.

Manage data masking templates

- **Edit a data masking template**

  To edit a data masking template, find the template on the **Desensitization Template** page and click **Edit** in the Actions column. In the **Edit** panel, modify the description or rules of the data masking template.



- **Delete a data masking template**

  To delete a data masking template that you no longer use, find the template on the **Desensitization Template** page and click **Delete** in the Actions column.

  > **Note** If you deleted a data masking template, it cannot be restored. Proceed with caution.

27.1.10.3.7.4. Configure data masking algorithms

This topic describes how to configure data masking algorithms.

Context

The following table describes the data masking algorithms that are supported by Sensitive Data Discovery and Protection (SDDP).

Category	Description	Algorithm	Input	Suitable sensitive data and scenario
Hashing	Raw data cannot be retrieved after it is masked. This type of algorithms is suitable for password masking and the scenarios in which you must check whether data is sensitive by comparison. You can use common hashing algorithms and specify a salt value.	MD5	Salt value	• Sensitive data: keys • Scenario: data storage
		SHA-1	Salt value	
		SHA-256	Salt value	
		HMAC	Salt value	

Redaction by using asterisks (*) or number signs (#)	Raw data cannot be retrieved after it is masked.
	This type of algorithms is suitable for the scenario in which you need to show sensitive data on a graphical user interface (GUI) or share sensitive data.
	This type of algorithms masks specific content in sensitive data by using asterisks (*) or number signs (#).
		Keeps the first N characters and the last M characters	Values of N and M	• Sensitive data: sensitive personal information
				• Scenarios:
				  ◦ Data usage
				  ◦ Data sharing

		Keeps characters from the Xth position to the Yth position	Values of X and Y	
		Masks the first N characters and the last M characters	Values of N and M	
		Masks characters from the Xth position to the Yth position	Values of X and Y	
		Masks characters that precede a special character when the special character appears for the first time	At sign (@), ampersand (&), or period (.)	

Category	Description	Algorithm	Input	Suitable sensitive data and scenario
		Masks characters that follow a special character when the special character appears for the first time	At sign (@), ampersand (&), or period (.)	
Substitution (customization supported)	Raw data can be retrieved after it is masked by using some of the algorithms. This type of algorithms can be used to mask fields in fixed formats. For example, you can use the algorithms to mask ID card numbers.	Substitutes specific content in ID card numbers with mapped values	Mapping table for randomly substituting IDs of administrative regions	• Sensitive data: ◦ Sensitive personal information ◦ Sensitive information of enterprises ◦ Sensitive information of devices • Scenarios: ◦ Data storage ◦ Data sharing
		Randomly substitutes specific content in ID card numbers	Mapping table for randomly substituting IDs of administrative regions	
		Randomly substitutes specific content in IDs of military officer cards	Mapping table for randomly substituting type codes	
		Randomly substitutes specified content in passport numbers	Mapping table for randomly substituting purpose fields	
		Randomly substitutes specific content in permit numbers of Mainland Travel Permits for Hong Kong and Macao Residents	Mapping table for randomly substituting purpose fields	
		Randomly substitutes specific content in bank card numbers	Mapping table for randomly substituting bank identification numbers (BINs)	
		Randomly substitutes specific content in landline telephone numbers	Mapping table for randomly substituting IDs of administrative regions	
		Randomly substitutes specific content in mobile phone numbers	Mapping table for randomly substituting mobile network codes	
		Randomly substitutes specific content in unified social credit codes	Mapping table for randomly substituting IDs of registration authorities, mapping table for randomly substituting type codes, and mapping table for randomly substituting IDs of administrative regions	

Category	Description	Algorithm	Input	Suitable sensitive data and scenario
	This type of algorithms substitutes the entire value or a part of the value of a field with a mapped value by using a mapping table. In this case, raw data can be retrieved after it is masked. This type of algorithms also substitutes the entire value or a part of the value of a field randomly based on a random interval. In this case, raw data cannot be retrieved after it is masked. SDDP provides multiple built-in mapping tables and allows you to customize substitution algorithms.	Substitutes specific content in general tables with mapped values.	Mapping table for substituting uppercase letters, mapping table for substituting lowercase letters, mapping table for substituting digits, and mapping table for substituting special characters	
		Randomly substitutes specific content in general tables	Mapping table for randomly substituting uppercase letters, mapping table for randomly substituting lowercase letters, mapping table for randomly substituting digits, and mapping table for randomly substituting special characters	

Rounding	Raw data can be retrieved after it is masked by using some of the algorithms.
	This type of algorithms can be used to analyze and collect statistics on sensitive datasets.
	SDDP provides two types of rounding algorithms. One algorithm rounds numbers and dates, and raw data cannot be retrieved after it is masked. The other algorithm bit-shifts text, and raw data can be retrieved after it is masked.
		Rounds down a number to the Nth digit before the decimal point	N	• Sensitive data: general sensitive information
				• Scenarios:
				  ◦ Data storage
				  ◦ Data usage

		Rounds dates	Date rounding level	
		Shifts characters	Number of places by which specific bits are moved and shift direction (left or right)	

Category	Description	Algorithm	Input	Suitable sensitive data and scenario

Encryption	Raw data can be retrieved after it is masked.
	This type of algorithms can be used to encrypt sensitive fields that need to be retrieved after encryption.
	Common symmetrical encryption algorithms are supported.
		Data Encryption Standard (DES) algorithm	Encryption key	• Sensitive data:
				  ◦ Sensitive personal information
				  ◦ Sensitive information of enterprises
				• Scenario: data storage

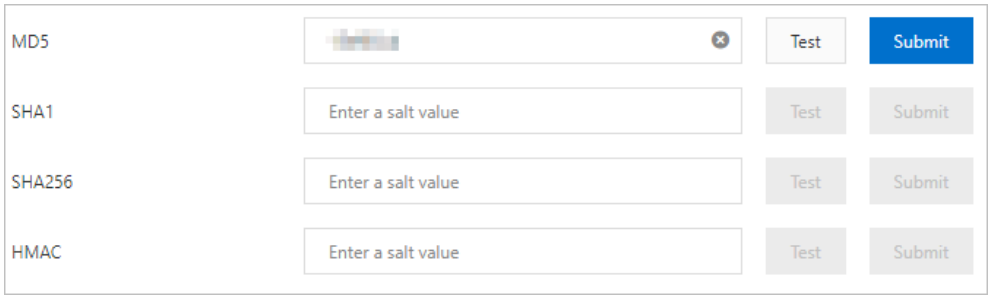
		Triple Data Encryption Standard (3DES) algorithm	Encryption key	
		Advanced Encryption Standard (AES) algorithm	Encryption key	

Shuffling	Raw data cannot be retrieved after it is masked.
	This type of algorithms can be used to mask structured data columns.
	This type of algorithms extracts values of a field in a specified range from the source table and rearranges the values in a specific column. Alternatively, this type of algorithms randomly selects values from a specific column within the value range and rearranges the selected values. This way, the values are mixed up and masked.
		Randomly shuffles data	Rearranged values or randomly selected values	• Sensitive data:
				  ◦ Sensitive information of devices
				  ◦ Location-sensitive information
				• Scenario: data storage

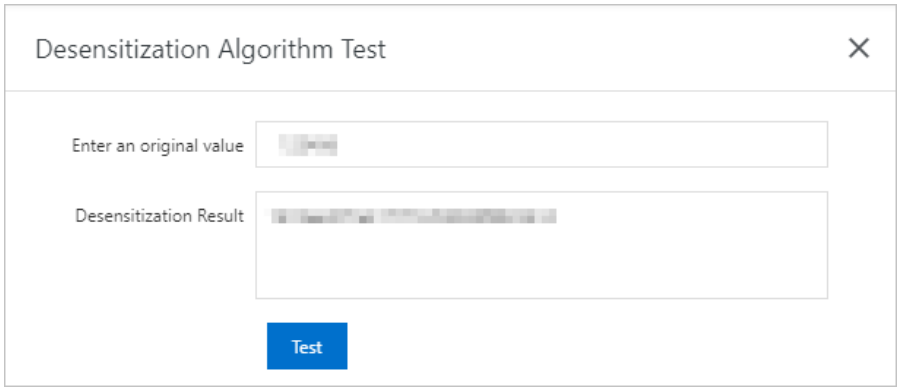
Hashing

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection.Sensitive Data DeidentificationDeidentification algorithms**.
3. Click the **Hashing** tab.
4. Specify a salt value for each algorithm.

> **Note** In cryptography, you can insert a specific string to a fixed position of a password to generate a hash value that is different from that of the original password. This process is called salting.
>
> A salt value is the specific string that you insert.



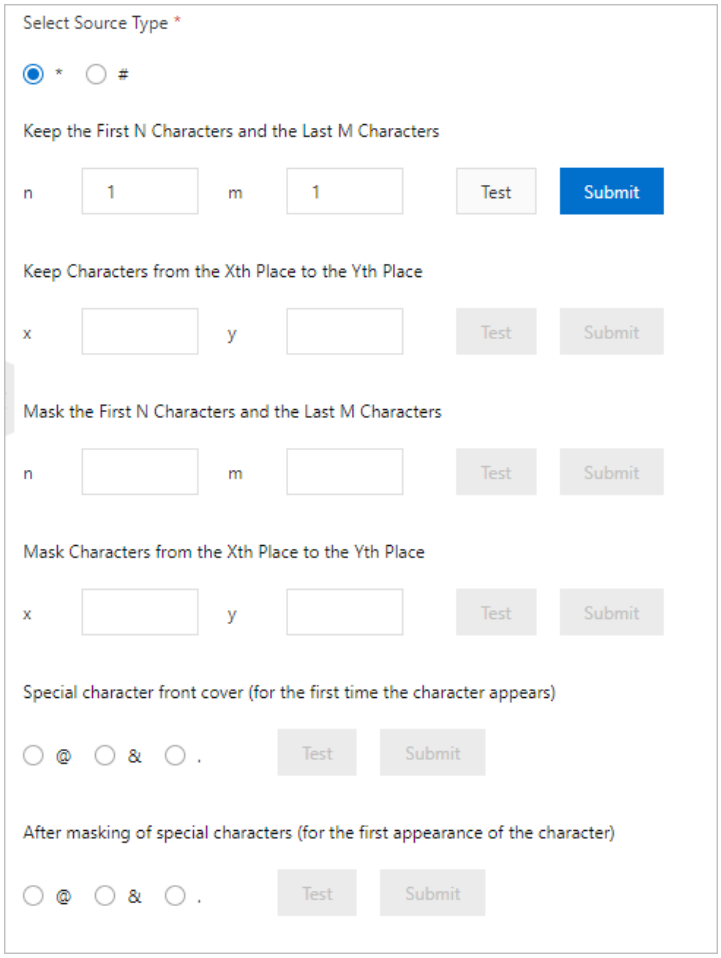
5. In the **Desensitization Algorithm Test** dialog box, enter the original value and click **Test** to check whether the algorithm works.



6. After the configuration is complete, click **Submit**.

Redaction

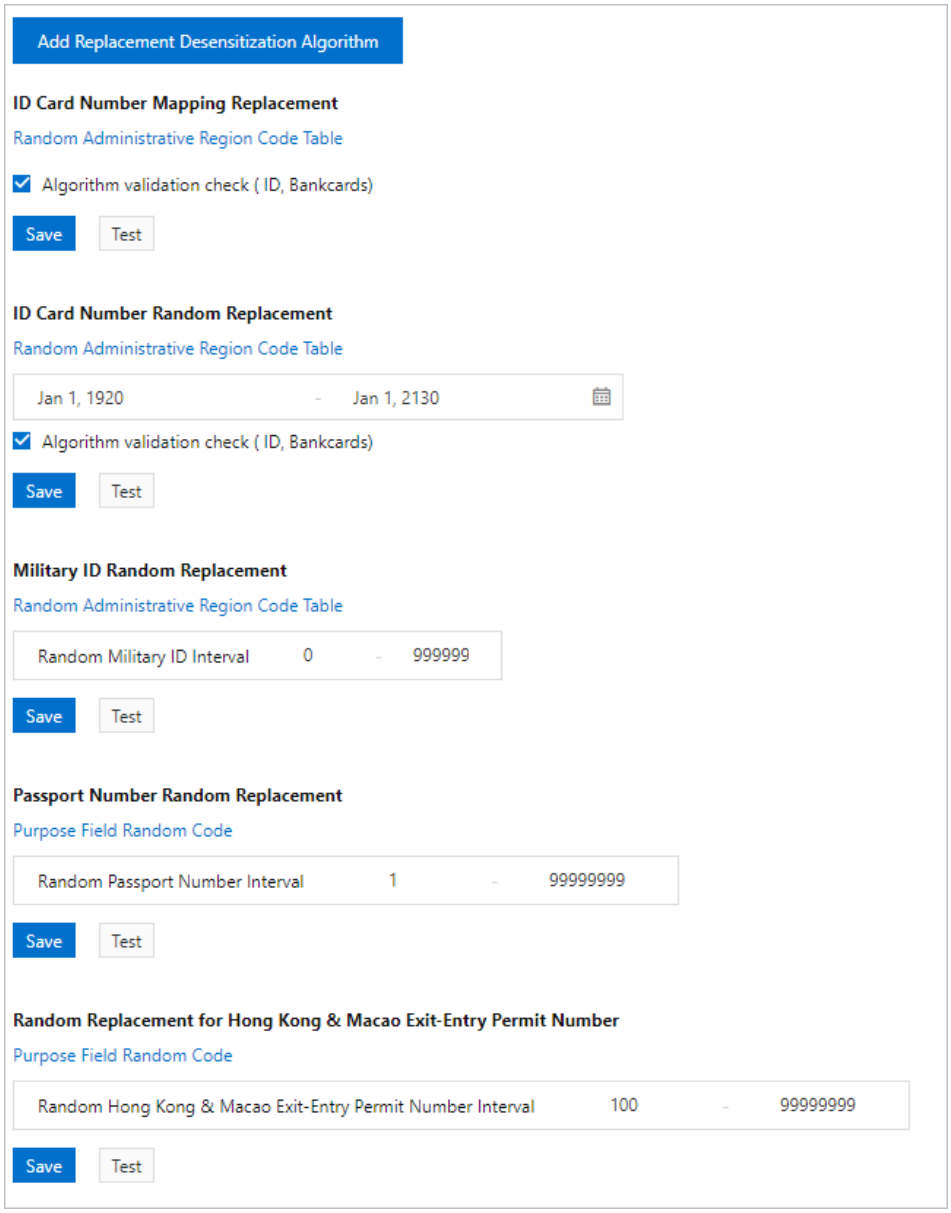
1.
2. Choose **Data Security > Sensitive Data Discovery and Protection**.
3. Click the **Masking** tab.
4. Configure the parameters.



5. In the **Desensitization Algorithm Test** dialog box, enter the original value and click **Test** to check whether the algorithm works.
6. After the configuration is complete, click **Submit**.

Substitution

1.
2. Choose **Data Security > Sensitive Data Discovery and Protection**.
3. Click the **Replacement** tab.
4. Configure the parameters.



Note By default, SDDP provides multiple common substitution algorithms, such as ID Card Number Mapping Replacement and Telephone Number Random Replacement.

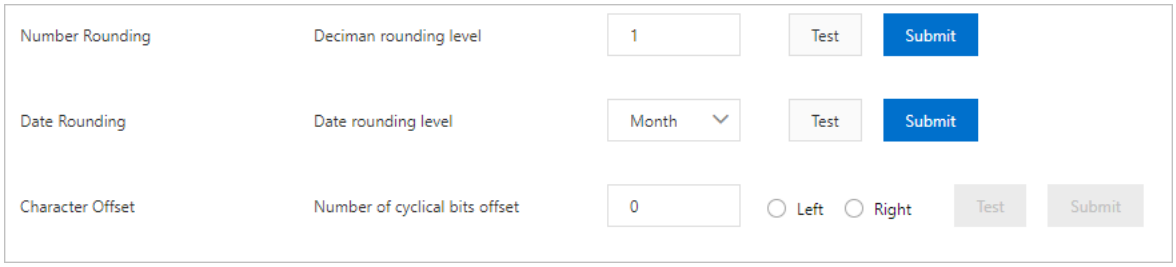
- If you want to customize a mapping table, click the required mapping table, replace the original content with your own mapping table, and then click **Save**.
- If you need to customize an algorithm, click **Add Replacement Desensitization Algorithm** and specify the interval and mapping table.

5. In the **Desensitization Algorithm Test** dialog box, enter the original value and click **Test** to check whether the algorithm works.
6. After the configuration is complete, click **Save**.

Rounding

1.
2. Choose **Data Security > Sensitive Data Discovery and Protection**.
3. Click the **Transformation** tab.

4. Configure the parameters.



5. In the **Desensitization Algorithm Test** dialog box, enter the original value and click **Test** to check whether the algorithm works.
6. After the configuration is complete, click **Submit**.

Encryption

1.
2. Choose **Data Security > Sensitive Data Discovery and Protection**.
3. Click the **Encryption** tab.
4. Specify a key for an algorithm.

5. In the **Desensitization Algorithm Test** dialog box, enter the original value and click **Test** to check whether the algorithm works.
6. After the configuration is complete, click **Submit**.

Shuffling

1.
2. Choose **Data Security > Sensitive Data Discovery and Protection**.
3. Click the **Shuffling** tab.
4. Select a shuffling method.

5. Click **Submit**.

27.1.10.3.7.5. Extract watermarks

You can add watermarks when you create a data masking task. If data is leaked after it is distributed, you can use watermarks to trace operations. This minimizes the impacts of data leaks. SDDP extracts and identifies watermarks from the leaked data to trace the data flow process and identify the organization or user that is responsible for the data leaks. This topic describes how to extract watermarks.

Procedure

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection**. In the left-side navigation tree, choose **Sensitive Data Desensitization** > **Extract watermark**.
3. On the **Extract watermark** page, configure the **Source Product**, **Source database/project name**, and **Source table name** parameters, and click **Extract watermark**.

Parameter	Description
Source Product	The name of the source data asset to which the table containing watermarks belongs.
Source database/project name	The name of the database or project to which the table containing watermarks belongs.
Source table name	The name of the table that contains watermarks.

   The extracted watermarks appear in the field below this parameter. If you want to copy the information, click **Copy Result**.

27.1.10.3.8. Data security lab

27.1.10.3.8.1. View data assets

The Data Asset map page displays statistics on your data assets and charts of sensitive data for the last seven days. This topic describes how to view data assets on the Data Asset map page.

Prerequisites

Sensitive Data Discovery and Protection (SDDP) is authorized to protect your data assets. For more information, see Authorize SDDP to access data assets.

View the statistics and charts of data assets

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data security lab > Data Asset map**.
3. On the Data Asset map page, click the tab of the data assets that you want to view. You can click **OSS**, **RDS**, **MaxCompute**, **ECS self-built database**, **DRDS**, **OTS**, **ADS**, or **GPDB**.
4. View the statistics and charts of your data assets.

   You can view the following statistics and charts:

   > **Note** In this example, the statistics and charts on the **OSS** tab are described. For more information about other data assets, log on to Apsara Stack Security Center.

   - **Number of buckets**: displays the total number of OSS buckets that are protected by SDDP. It also displays the total number of OSS buckets that contain sensitive data.
   - **Number of files**: displays the total number of OSS objects that are protected by SDDP. It also displays the total number of OSS objects that contain sensitive data.
   - **Hit Rule TOP10**: displays the top 10 sensitive data detection rules that are hit.
   - **Proportion of sensitive files**: displays the proportions of public, less sensitive, private, and confidential objects in OSS by using a pie chart.

- **Data volume change**: displays the change curve of the total number of OSS objects that are protected by SDDP. It also displays the total number of OSS objects that contain sensitive data.

View the detailed information about the data assets

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data security lab > Data Asset map**.
3. On the Data Asset map page, click the tab of the data assets that you want to view. You can click **OSS**, **RDS**, **MaxCompute**, **ECS self-built database**, **DRDS**, **OTS**, **ADS**, or **GPDB**.
4. View the detailed information about the data assets.

   You can perform the following operations to view the detailed information about the data assets:

   > **Note** In this example, the detailed information about the data assets on the **OSS** tab is described. For more information about other data assets, log on to Apsara Stack Security Center.

   - View buckets. You can view the information about buckets at highly sensitive, moderately sensitive, low sensitive, and non-sensitive levels. The information includes regions where buckets reside, bucket names, the total number of objects, data security scores, the number of sensitive objects, and the last scan time of each bucket.
   - View details of an object. Find a specific bucket and click **Details** in the Actions column. On the OSS page, you can view the proportion of sensitive objects in the bucket, the top five sensitive data detection rules that are hit, and file details list. In the file details list, find a specific object and click **Hit details** in the Actions column. Then, you can view the information about the sensitive data detection rules that are hit.
   - View an asset portrait of a bucket. Find a specific object and click **Asset portrait** in the Actions column. On the Asset portrait page, you can view the security score of the bucket, number of objects, number of accounts, number of IP addresses, change curve of the security scores, anomalous events, and alert details.

27.1.10.3.8.2. View account information

The user account analysis feature provides the statistics and security scores of the last seven days for all accounts in your assets. This gives you an overview of your accounts and the security status. This topic describes how to use the user account analysis feature.

Procedure

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data security lab > User account analysis**.
3. On the **User account analysis** page, perform the following steps:
   - View account statistics: In the upper part of the page, you can view the statistics in the **Total Users**, **Dormant users**, **Active users**, and **High-risk users** sections. You can also view the **Account number change** line chart.
   - View account details: In the account list, you can filter accounts by account type and view the information of the accounts. The account types are Main account, RAM account number, and RDS account. The information includes Instance, Account security point, Account creation time, First occurrence time, Authorized assets, and Number of alarms. In the Actions column of an account, you can click **Account portrait** to view the profile details of the account. The profile details include the security score, user information, trend in the security score, trend in the number of anomalous events, and alert details.

> **Notice** If the security score of an account is low, it indicates that the account may have performed unauthorized operations, such as accessing or downloading sensitive data. We recommend that you view the reasons that cause a low security score on the **Account portrait** page of the account, and handle the alerts and anomalous events.

27.1.10.3.8.3. Handle anomalous events

Sensitive Data Discovery and Protection (SDDP) detects anomalous events that are related to sensitive data and generates alerts. The SDDP console displays statistics about different types of anomalous events. You can process detected anomalous events in a centralized manner.

Background information

SDDP classifies anomalous events into the following types:

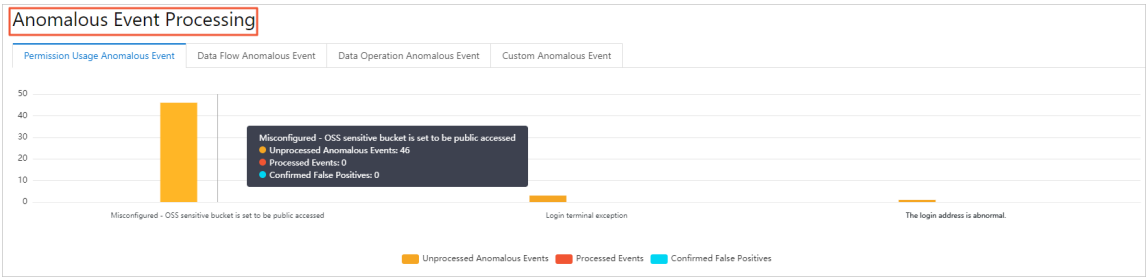
- **Permission Usage Anomalous Event**: Permissions are used in an inappropriate manner. For example, a user logs on from a disapproved IP address or by using the AccessKey pair of another user.
- **Data Flow Anomalous Event**: Anomalous events are detected in data flows. For example, a user unnecessarily downloads sensitive data files or downloads sensitive data files during an unusual period of time.
- **Data Operation Anomalous Event**: Unusual operations are performed on sensitive data. For example, a user modifies sensitive fields.
- **Custom Anomalous Event**: Anomalous events are detected and alerts are triggered based on custom rules. For more information, see Create a custom rule.

Step 1: View anomalous events

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data security lab > Data Asset map**.
3. On the **Anomalous Event Processing** page, view statistics about anomalous events and the anomalous event list.

   You can click the **Permission Usage Anomalous Event**, **Data Flow Anomalous Event**, **Data Operation Anomalous Event**, or **Custom Anomalous Event** tab to view the information about related events.

   - In the upper part of the **Anomalous Event Processing** page, you can view statistics about anomalous events. The statistics include the types of anomalous events, number of unhandled anomalous events, number of handled anomalous events, and number of anomalous events that are marked as false positives.

   - In the lower part of the **Anomalous Event Processing** page, you can view the list of anomalous events that are detected. You can filter anomalous events by **Event Type**, **Status**, and time range. An anomalous event can be in one of the following states: To be processed, Processed, and False positive. A time range specifies the period of time during which alerts are generated for anomalous events.

     > **Note** You can specify a time range based on your business requirements.

Step 2: Handle anomalous events

In the list of anomalous events on the **Anomalous Event Processing** page, you can view the details about anomalous events and handle the anomalous events.

- To view the details about an anomalous event, click **View Details** in the Actions column of the anomalous event. On the Anomalous event details page, you can view the basic information about the anomalous event, the service where the anomalous event occurs, event description, and suggestions on how to handle the event.
- To handle an anomalous event, click **Process** in the Actions column of the anomalous event. On the **Exception event handling** page, handle the event.

  In most cases, SDDP reports an anomalous event if an unauthorized user accesses or downloads sensitive data, an authorized user accesses or downloads sensitive data in an unusual period of time, or a user accesses sensitive data from an unusual terminal. After SDDP detects anomalous events, it displays them on the Exception event handling page.

  You can handle anomalous events on the **Exception event handling** page. Configure the following parameters:

  - **Anomalous Event Verification**
    - **Confirmed and Processed**: If you verify that a detected event is an anomalous event, select this option. In this case, find the event based on the information provided on the **Anomalous event details** page. Follow the instructions in the **Event Processing Suggestion** section of the page and manually handle the event in the data source of the event. If you select this option but do not handle the event, SDDP continues to generate alerts for the event.
    - **False Positive**: If you verify that a detected event is a false positive and can be ignored, select this option. After you select this option, SDDP no longer generates alerts for this event. This event no longer appears on the Anomalous Events page.
  - **Processing method**: Turn on **Set Bucket Acl to private**. The  icon indicates that the bucket access control list (ACL) is set to private. In this case, you must use AccessKey pairs to access the bucket. You can click **Ban history** to access the **Ban history** page, view the buckets whose ACL is set to private, and change the ACL of a bucket.
  - **Add Processing Record**: Enter a description for the anomalous event to facilitate subsequent analysis.
  - **Sample-based enhancement for anomalous events**: If you select **False Positive** in the **Anomalous Event Verification** section and select this option, SDDP adds the anomalous event to a library of false positive samples. The library helps SDDP accurately detect anomalous events.

27.1.10.3.8.4. Create a custom rule

Sensitive Data Discovery and Protection (SDDP) allows you to create custom rules to detect anomalous events and generates alerts for the events. SDDP captures log data of the anomalous events that match the custom rules. This topic describes how to create a custom rule to detect anomalous events.

Context

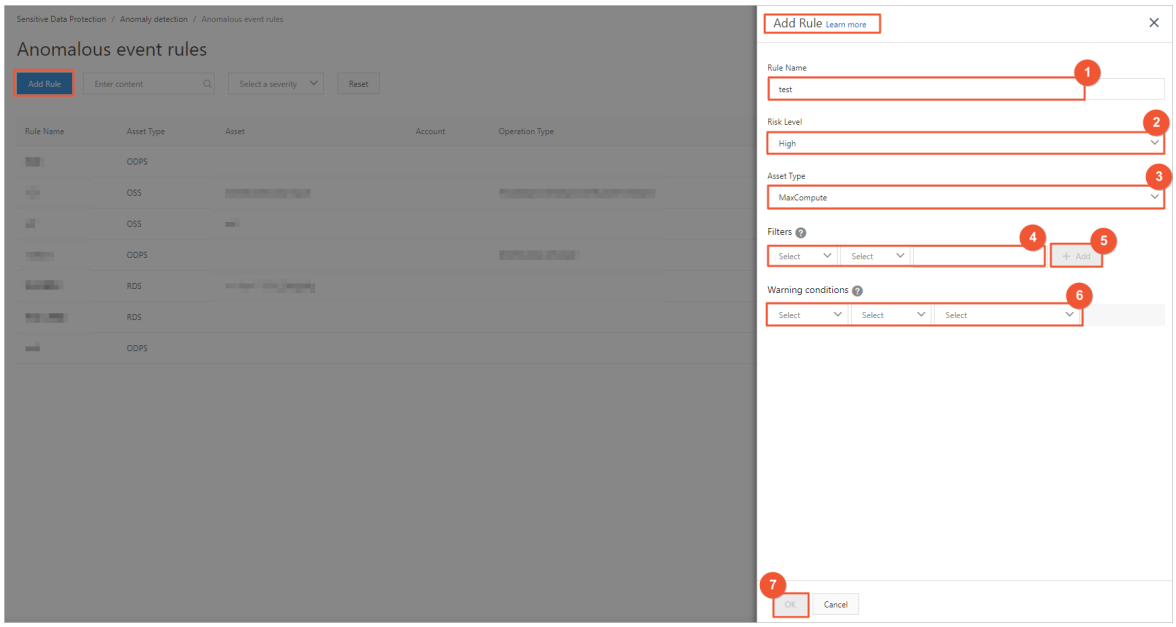
SDDP can detect anomalous events and generate alerts based on both built-in and custom rules.

> **Note** Custom rules are supported only for the following data sources: Object Storage Service (OSS), MaxCompute, and ApsaraDB RDS. Only built-in rules are supported for other data sources.

Procedure

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data security lab > Custom rules**.
3. Click **Add Rule**.

4. In the **Add Rule** panel, configure the parameters to create a custom rule.



The following table describes the parameters.

Parameter	Description
Rule Name	The name of the custom rule to detect anomalous events. We recommend that you enter an informative name to identify the rule.
Risk Level	The risk level of the anomalous events to be detected. Valid values: ○ **High** ○ **Medium** ○ **Low**
Asset Type	The type of the asset for which SDDP checks for anomalous events based on the rule. Valid values: ○ **OSS** ○ **MaxCompute** ○ **RDS**
Filters	The filter that specifies the anomalous events to be detected. The filters of a rule have logical AND relations. Click **Add** to add a filter. You can add multiple filters.

Parameter	Description
Warning Conditions	The time interval to detect anomalous events and the alert triggering conditions. SDDP detects anomalous events based on the Filters parameter that you specify. If the specified alert triggering conditions are met in the specified time interval, SDDP generates alerts for the events. **Note** The **Any UA** option in Warning Conditions specifies that the User-Agent information of all types of browsers is detected. User-Agent information includes the hardware platform, system software, application software, and user preferences. You can obtain the following information based on User-Agent information: the name, version number, and rendering engine of the browser, and the operating system on which the browser runs. If you select **Any UA** in Warning Conditions, SDDP detects for anomalous events based on User-Agent information.

5. Click **OK**.

   After you create the custom rule, you can view the enabling status, matching results, and details of the rule on the **Custom rules** page. You can also edit the rule.

   Note A custom rule automatically enters the **Enabled** state after it is created. You can also manually disable the rule on the **Custom rules** page. After you disable a rule, SDDP no longer detects for anomalous events based on the rule.

What's next

After you create and enable a custom rule, SDDP provides the detection result based on this rule on the **Exception event handling** page in about 1 hour. You can view the result on this page. For more information, see Handle anomalous events.

27.1.10.3.8.5. Configure alerts

Sensitive Data Discovery and Protection (SDDP) detects anomalous event in your data assets and generates alerts based on built-in and custom anomalous event detection rules. On the Built-in detection model page, you can configure the general settings for alerts and enable alerts for different types of anomalous event based on your business requirements. This topic describes how to configure alerts.

Configure the general settings for alerts

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection**. In the left-side navigation tree, choose **Data security lab > Built-in rules**.
3. On the **Built-in detection model** page, find the required item in the **General Configuration for Anomaly Alerts** section and click **Modify** next to the item.
4. Modify the threshold to trigger an alert.

   SDDP allows you to set thresholds for the following types of anomalous events:

   - **Data access inactivity**: After you enable alerts for this anomalous event in the **Enable Anomaly Alerts** section, SDDP generates an alert when the idle period of a permission exceeds the specified threshold. You can view the detailed alert information on the **Anomalous Events** page.
   - **Log output anomaly detected**: After you enable alerts for this anomalous event in the **Enable Anomaly Alerts** section, SDDP generates an alert when the amount of logs that are generated in a day is lower than the specified threshold. You can view the detailed alert information on the **Anomalous Events**

page.

- **Unauthorized resource access attempts**: After you enable alerts for this anomalous event in the **Enable Anomaly Alerts** section, SDDP generates an alert when the number of access attempts exceeds the specified threshold. You can view the detailed alert information on the **Anomalous Events** page. For more information, see Handle anomalous events.

5. Click **Submit**.

Enable alerts for anomalous event

1.
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection**. In the left-side navigation tree, choose **Data security lab > Built-in rules**.
3. On the **Built-in detection model** page, select the required anomalous event for which you want to enable alerts in the **Enable Anomaly Alerts** section.

   After alerts are generated for the selected anomalous event, you can view the detailed alert information on the **Anomalous Events** page. For more information, see Handle anomalous events.

28.Log Service

28.1. User Guide

28.1.1. What is Log Service?

Log Service (SLS) is a one-stop logging service developed by Alibaba Cloud that is widely used by Alibaba Group in big data scenarios. You can use Log Service to collect, query, and consume log data.

Without the need to invest in in-house data collection and processing systems. This enables you to focus on your business, improving business efficiency and helping your business to expand.

Log Service provides the following features:

- Log collection: Log Service allows you to collect events, binary logs, and text logs in real time by using multiple methods, such as Logtail and JavaScript.
- Query and analysis: Log Service allows you to query and analyze the collected log data and view analysis results on charts and dashboards.
- Status alert: Log Service can automatically run query statements at regular intervals after you create an alert task. If the query results meet the conditions of the alert task, Log Service sends an alert to the specified recipients in real time.
- Real-time consumption: Log Service provides real-time consumption interfaces through which log consumers can consume log data.

28.1.2. Quick start

28.1.2.1. Procedure

This topic provides the basic procedure to use Log Service. You can use this procedure to create projects, create Logstores, and collect log data.

1. Optional. Obtain an AccessKey pair.

   Before you can use Log Service through APIs or SDKs, you must have an AccessKey pair.

2. Create a project.

   Create a project in a specified region and add a description.

3. Create a Logstore.

   Create a Logstore for the project and specify the number of shards.

4. Collect text logs.

   Select a method to collect log data based on your business requirements. Text log collection is used as an example.

5. Enable the index feature and configure indexes for a Logstore, and query and analyze logs.

   Log Service supports real-time log query and analysis. After you enable the indexing feature, you can query and analyze logs and configure Overview and dashboards.

6. Configure alerts.

   Log Service allows you to configure alerts based on log query results. Then, Log Service sends alerts by using multiple methods, such as a custom webhook.

7. Consume logs in real time.

   Log Service allows you to consume logs by using multiple methods, such as a Spark Streaming client, Storm spout, and Flink connector.

28.1.2.2. Log on to the Log Service console

This topic describes how to log on to the Log Service console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products > Log Service**.
5. On the page that appears, select the organization and region, and then click **SLS**. The home page of the Log Service console is displayed.

28.1.2.3. Obtain an AccessKey pair

An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey pair is used to implement symmetric encryption to verify the identity of the requester. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt the signature string. This topic describes how to obtain an AccessKey pair.

Prerequisites

Only the operation administrators or level-1 organization administrators can obtain the AccessKey pair of an organization.

Context

To call Apsara Uni-manager and cloud service APIs, we recommend that you use the AccessKey pair of a personal account. If you use the AccessKey pair of a personal account, you must configure header parameters as described in the following table for access control.

Parameter	Description
x-acs-regionid	The region ID, such as cn-hangzhou-*.
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console.

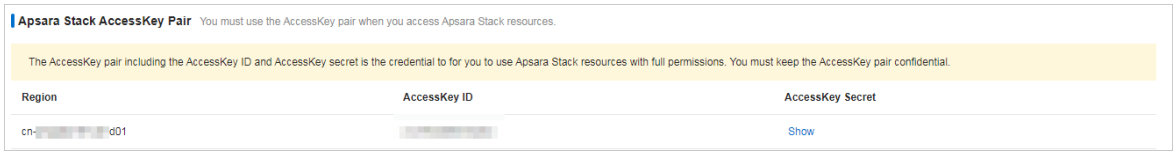
Parameter	Description
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console.
x-acs-instanceid	The ID of the instance on which you want to perform operations.

Warning The AccessKey pairs of personal accounts are under control of the Apsara Uni-manager permission system. AccessKey pairs of organization accounts have higher permissions. For security purposes, organization operations must be approved by administrators.

Obtain the AccessKey pair of a personal account

To obtain the AccessKey pair of a personal account, perform the following operations:

1. Log on to the Apsara Uni-manager Management Console.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **Personal Information**.
3. In the **Apsara Stack AccessKey Pair** section, view your AccessKey pair.



Note The AccessKey pair consists of an AccessKey ID and an AccessKey secret. AccessKey pairs allow you to access Apsara Stack resources with full permissions for your account. You must keep your AccessKey pair confidential.

Obtain the AccessKey pair of an organization

To obtain the AccessKey pair of an organization, perform the following operations:

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the organization navigation tree, click an organization name.
5. In the Current Organization section, click **Management Accesskey**.
6. In the Management AccessKey, view the AccessKey pair of the organization.

Note An AccessKey pair is automatically allocated to each level-1 organization. Subordinate organizations use the same AccessKey pair of their level-1 organization.

28.1.2.4. Manage projects

This topic describes how to create, modify, and delete projects in the Log Service console.

Context

A project in Log Service is a resource management unit. The resources in each project are isolated from resources in other projects. We recommend that you store the log data of different applications in dedicated projects. You can manage Logstores, Logtail configurations, log sources, log data, and machine groups in a project. Each

project provides an endpoint for you to access the resources.

A project provides the following features:

- Allows you to store log data from different sources in different Logstores of a project. You can use Log Service to collect log data from multiple sources such as business projects, products, and environments. You can then store the log data of each source in a separate Logstore. This simplifies the downstream processes such as the consuming, exporting, and indexing of log data. In addition, you can manage access permissions at the project level.
- Provides an endpoint for you to access the resources in the project. Log Service allocates an exclusive endpoint to each project. You can use the endpoint to read, write, and manage the log data in the project.

Create a project

> **Note**
> - You can create a project only by using the Log Service console.
> - You can create up to 50 projects under each Apsara Stack tenant account.

1. Log on to the Log Service console.
2. In the **Projects** section, click **Create Project**.
3. Set the parameters based on your requirements. The following table describes the parameters.

Parameter	Description
Project Name	The name of the project. The project name must be unique across all regions. Use the following naming conventions: ○ The name can contain lowercase letters, digits, and hyphens (-). ○ The name must start and end with a lowercase letter or digit. ○ The name must be 3 to 63 characters in length. **Note** After a project is created, its name cannot be modified.
Description	The description of the project. After the project is created, the description is displayed in the **Projects** section. If you need to modify the description after the project is created, find the project in the **Projects** section, and click **Edit** in the **Actions** column. The description must be 0 to 64 characters in length and cannot contain the following characters: `< > ' \ " \\` .
Region	The region to which the project belongs. We recommend that you select a region that is closer to the log source. After a project is created, its region cannot be modified. This means that projects cannot be migrated across regions.

4. Click **OK**.

Modify the description of a project

To modify the description of a project, perform the following steps:

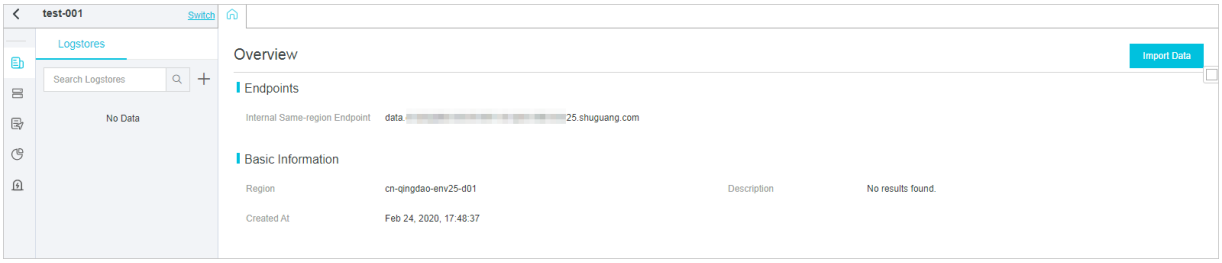
1. In the **Projects** section, find the project.
2. In the **Actions** column, click **Edit**.
3. In the **Modify Project** dialog box, modify the description of the project.

> **Note** You cannot modify the project name or region.

4. Click **OK**.

View the information of a project

To view the information of a project, click the project name in the **Projects** section. On the **Overview** page, you can view the project information such as the endpoint and region.



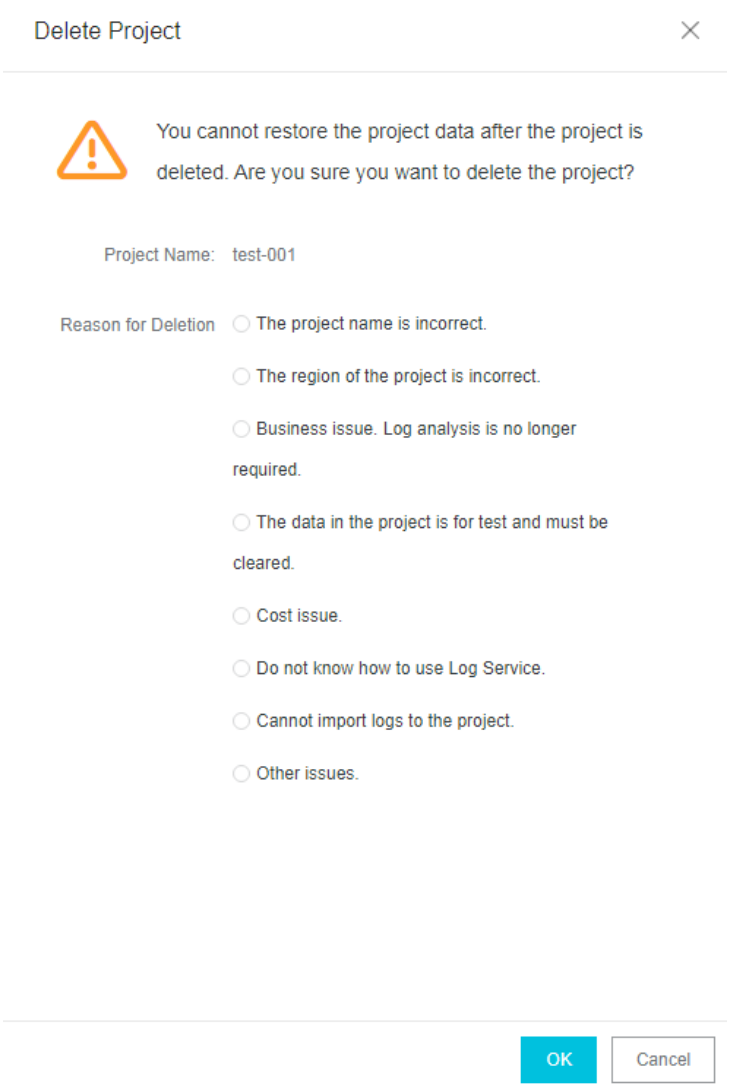
Delete a project

To delete a project, perform the following steps:

> **Warning** After you delete a project, all logs and configurations in the project are deleted and cannot be restored.

1. In the **Projects** section, find the project.
2. In the **Actions** column, click **Delete**.
3. In the dialog box that appears, select a reason for deletion.

   If you select **Other issues**, enter the reason in the text box.



4. Click **OK**.

28.1.2.5. Manage Logstores

This topic describes how to create, modify, and delete a Logstore in the Log Service console. A Logstore is a collection of resources inside a project. The log data in a Logstore is collected from the same source.

Context

You can create multiple Logstores in a project. We recommend that you create a Logstore for each type of application log.

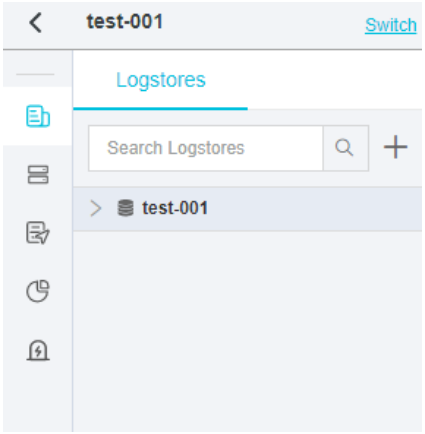
Logstores provide the following features:

- Real-time log collection
- Log storage and real-time log consumption
- Log indexing and real-time log query

Create a Logstore

> **Note** You can create a maximum of 100 Logstores in each project.

1. Log on to the Log Service console.
2. In the **Projects** section, click the name of a project.
3. On the page that appears, click next to the search box.



4. In the **Create Logstore** pane, set the parameters and click **OK**.

Parameter	Description
Logstore Name	The name of the Logstore. The Logstore name must be unique in the project to which the Logstore belongs. ○ The name can contain lowercase letters, digits, hyphens (-), and underscores (_). ○ The name must start and end with a lowercase letter or digit. ○ The name must be 3 to 63 characters in length. **Note** After a Logstore is created, its name cannot be modified.
WebTracking	Specifies whether to enable the WebTracking feature for the Logstore. You can use WebTracking to collect the log data of HTML websites, HTML5 websites, iOS apps, or Android apps and forward the data to Log Service. This feature is not enabled by default.
Permanent Storage	Specifies whether to permanently store the log data in the Logstore. If you disable this feature, you must specify a retention period for log data.
Data Retention Period	The duration for which log data is stored in the Logstore after the log data is collected. Unit: days. Valid values: 1 to 3000. When this period expires, the log data is deleted.
Shards	The number of shards in the Logstore. You can divide a Logstore into 1 to 10 shards.
Automatic Sharding	Specifies whether to enable the automatic sharding feature. This feature is not enabled by default. If you enable this feature, Log Service automatically splits shards in the Logstore when the data transfer exceeds the capacity of the existing shards.
Maximum Shards	The maximum number of shards. This parameter is required if you enable the automatic sharding feature. Maximum value: 64.

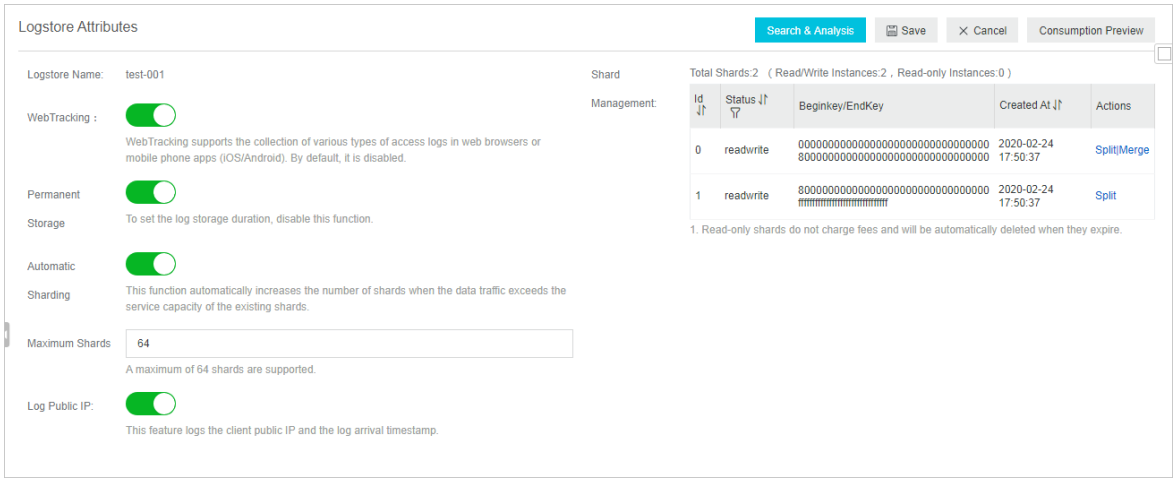
Parameter	Description
Log Public IP	Specifies whether to add the following information to the Tag field of each collected log entry: ○ `__client_ip__`: the public IP address of the log source. ○ `__receive_time__`: the time when Log Service receives the log entry. The value is a UNIX timestamp.

Modify the configurations of a Logstore

To modify the configurations of a Logstore, perform the following steps:

1. In the **Projects** section, click the name of the project to which the Logstore belongs.
2. On the page that appears, click the icon next to the Logstore, and select **Modify** from the shortcut menu.
3. In the upper-right corner of the **Logstore Attributes** page, click **Modify**.

   You can modify the **Data Retention Period**, **WebTracking**, **Automatic Sharding**, **Maximum Shards**, and **Log Public IP** parameters. You can also split or merge the existing shards.



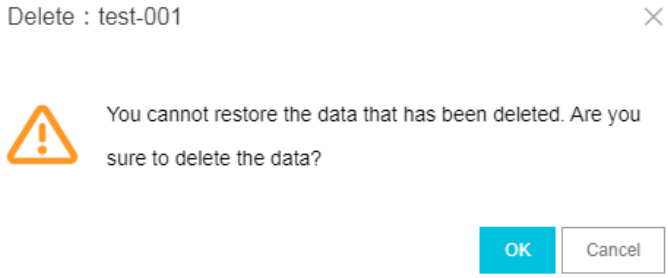
4. Click **Save**.

Delete a Logstore

To delete a Logstore, perform the following steps:

> **Note**
> - After you delete a Logstore, the log data in the Logstore is deleted and cannot be restored.
> - Before you can delete a Logstore, you must delete the Logtail configurations that are associated with the Logstore.

1. In the **Projects** section, click the name of the project to which the Logstore belongs.
2. On the page that appears, click the icon next to the Logstore, and select **Delete** from the shortcut menu.
3. In the message that appears, click **OK**.



28.1.2.6. Manage shards

This topic describes how to split, merge, and delete shards in the Log Service console. Logs are stored on shards in a Logstore. Each Logstore can have multiple shards. When you create a Logstore, you must specify the number of shards in the Logstore. After a Logstore is created, you can split or merge the shards.

Hash key

Log Service uses 128-bit MD5 hashes as the hash key of a Logstore. The entire MD5 hash range is [00000000000000000000000000000000,ffffffffffffffffffffffffffffffff). The hash key range of a Logstore falls within the entire MD5 hash range. When you create a Logstore, you must specify the number (N) of shards in the Logstore. The hash key range of the Logstore is evenly divided into N parts. Each part is assigned to a shard.

The hash key range of a shard is a left-closed and right-open interval that is specified by the following parameters:

- BeginKey: the start of the hash key range. The value of this parameter is included in the range.
- EndKey: the end of the hash key range. The value of this parameter is excluded from the range.

If you split a shard, the hash key range of the shard is evenly split. If you merge two shards, the hash key ranges of the shards are also merged. A hash key range determines the scope of a shard. When you push log data to a Logstore, you can specify a hash key for the log data. Log Service then writes the log data to the shard whose hash key range includes the specified hash key. This is called the hash key mode. If you do not specify a hash key for log data, the load balancing mode is used and Log Service writes the log data to a random available shard. However, when you pull log data from a Logstore, you must specify the shard where the log data is stored.

For example, a Logstore is divided into four shards and the hash key range of the Logstore is [00,FF). Example shards lists the hash key range of each shard.

Example shards

Shard	Hash key range
Shard0	[00,40)
Shard1	[40,80)
Shard2	[80,C0)
Shard3	[C0,FF)

If you set the hash key of log data to 5F, Log Service writes the log data to shard 1 because the hash key range of shard 1 includes 5F. If you set the hash key to 8C, the log data is written to shard 2 because the hash key range of shard 2 includes 8C.

Read/write capacity

Each shard provides an identical read/write capacity. Therefore, the read/write capacity of a Logstore depends on the number of shards in the Logstore. We recommend that you adjust the capacity of a Logstore based on the data traffic. For a Logstore, if the data traffic exceeds the read/write capacity, you can split shards to increase the Logstore capacity. If the data traffic is much less than the read/write capacity, you can merge shards to reduce the Logstore capacity and save costs.

For example, a Logstore consists of two read/write shards and the shards provide a maximum write capacity of 10 MB/s. If log data is written to the Logstore at a rate of 14 MB/s, we recommend that you split one of the shards into two shards. However, if log data is written at a rate of 3 MB/s, you can merge the two shards because the capacity of one shard already meets the read/write requirements.

> Note
> - If an API operation that writes data to a Logstore constantly returns 403 or 500 errors, you can check the data traffic metrics that are provided by Log Service and determine whether to split shards.
> - If the data traffic of a Logstore exceeds the read/write capacity of the Logstore, Log Service provides the best possible service but does not guarantee the service quality.

Shard status

A shard can be in one of the following states:

- Read/write
- Read-only

After a shard is created, the default status of the shard is read/write. If you split or merge shards, the status of the original shards changes to read-only and the new shards are in the read/write state. You can write data to and read data from a read/write shard. However, you can only read data from a read-only shard and cannot write data to the shard.

If you need to split a shard in a Logstore, you must specify the ID of the shard and an MD5 hash. The shard must be in the read/write state. The MD5 hash must be greater than the value of the BeginKey parameter of the shard and less than the value of the EndKey parameter of the shard. After the shard is split, the Logstore has two more shards. The status of the original shard changes from read/write to read-only. You can consume the log data in the original shard but cannot write log data to the shard. The new shards are in the read/write state and are listed below the original shard. The hash key ranges of the new shards cover that of the original shard.

If you need to merge shards in a Logstore, you must specify a read/write shard. The shard cannot be the last read/write shard in the shard list. Log Service finds the shard whose hash key range follows the hash key range of the specified shard, and merges the two shards into a new shard. The status of the original shards changes from read/write to read-only. You can consume the log data in the original shards but cannot write log data to the shards. The new shard is in the read/write state. The hash key range of the new shard covers those of the original shards.

You can perform the following operations on shards in the Log Service console:

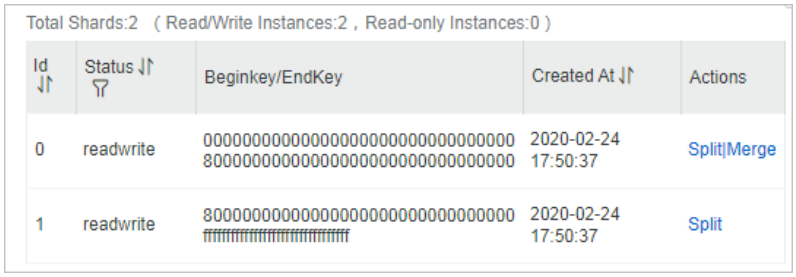
- Split a shard.
- Merge shards.

Split a shard

Each shard provides a write capacity of 5 MB/s and a read capacity of 10 MB/s. For a Logstore, if the data traffic exceeds the total read/write capacity of existing shards, we recommend that you split shards to increase the capacity.

1. Log on to the Log Service console.
2. In the Projects section, click the name of the project in which you want to create a Logstore.
3. In the left-side navigation pane, choose **Log Storage > Logstores**. On the Logstores tab, find the Logstore that you want to modify, click the  icon, and then select **Modify**.

4. In the upper-right corner of the **Logstore Attributes** page, click **Modify**.
5. In the **Shard Management** section, find the shard that you want to split, and click **Split** in the **Actions** column.



6. In the drop-down list, click the number of new shards.
7. In the **Actions** column, click **OK**.
8. In the upper-right corner of the **Logstore Attributes** page, click **Save**.

Enable the automatic sharding feature

Log Service provides the automatic sharding feature.

If you enable the automatic sharding feature for a Logstore, shards are automatically split if the following conditions are met:

- The data traffic of the Logstore exceeds the total capacity of the shards in the Logstore for more than five minutes.
- The number of read/write shards does not exceed the maximum number of shards that is allowed in the Logstore after an automatic shard split.

> **Note** After a shard is split into multiple shards, the new shards will not be automatically split within 15 minutes.

When you create or modify a Logstore, you can enable the automatic sharding feature. If this feature is enabled, you must specify the maximum number of shards that is allowed in the Logstore after an automatic shard split. For more information, see Manage Logstores.

- Automatic Sharding

  Specifies whether to enable the automatic sharding feature. If you turn on this switch, the shards are automatically split when the preceding conditions are met.

- Maximum Shards

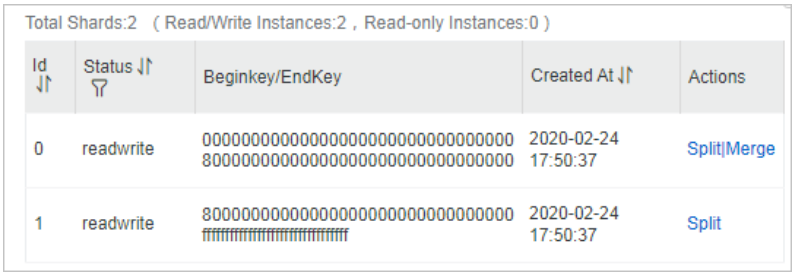
  The maximum number of shards that is allowed in the Logstore after an automatic shard split. The maximum value is 64.

Merge shards

You can merge shards in a Logstore to reduce the capacity of the Logstore.

1. In the upper-right corner of the **Logstore Attributes** page, click **Modify**.
2. In the **Shard Management** section, find the shard that you want to merge with another shard, and click

Merge in the **Actions** column.



3. In the upper-right corner of the **Logstore Attributes** page, click **Save**.

Delete a shard

- Automatic deletion

  When you create a Logstore, you can set a retention period for log data. When the retention period expires, the shards in the Logstore and the log data in the shards are automatically deleted.

- Manual deletion

  If you turn on the **Permanent Storage** switch when you create a Logstore, the shards in the Logstore and the log data in the shards are not automatically deleted. If you need to delete the shards and the log data in the shards, you can delete the Logstore. For more information, see Delete a Logstore.

28.1.3. Data collection

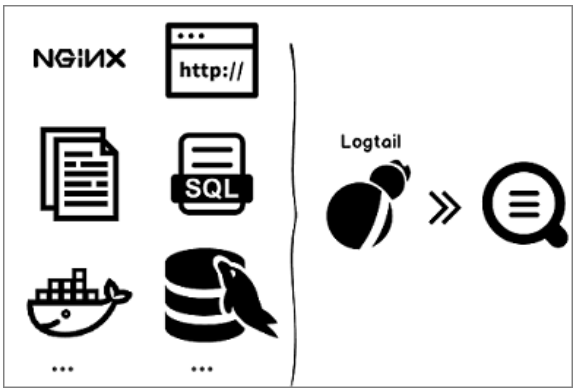
28.1.3.1. Collection by Logtail

28.1.3.1.1. Overview

28.1.3.1.1.1. Logtail overview

Logtail is a log collection agent provided by Log Service. You can use Logtail in Log Service console to collect logs in real time from multiple sources. These sources include Alibaba Cloud Elastic Compute Service (ECS) instances, servers deployed in on-premises data centers, and servers provided by other cloud service providers.

Logtail-based log collection



Benefits

- Provides non-intrusive log collection based on log files. You do not need to modify the application code. Log collection does not affect the operation logic of your applications.
- Supports the collection of multiple logs, such as text logs, binlogs, HTTP logs, and container stdout logs.
- Collects data from standard containers and various container clusters, such as swarm clusters and Kubernetes

clusters.

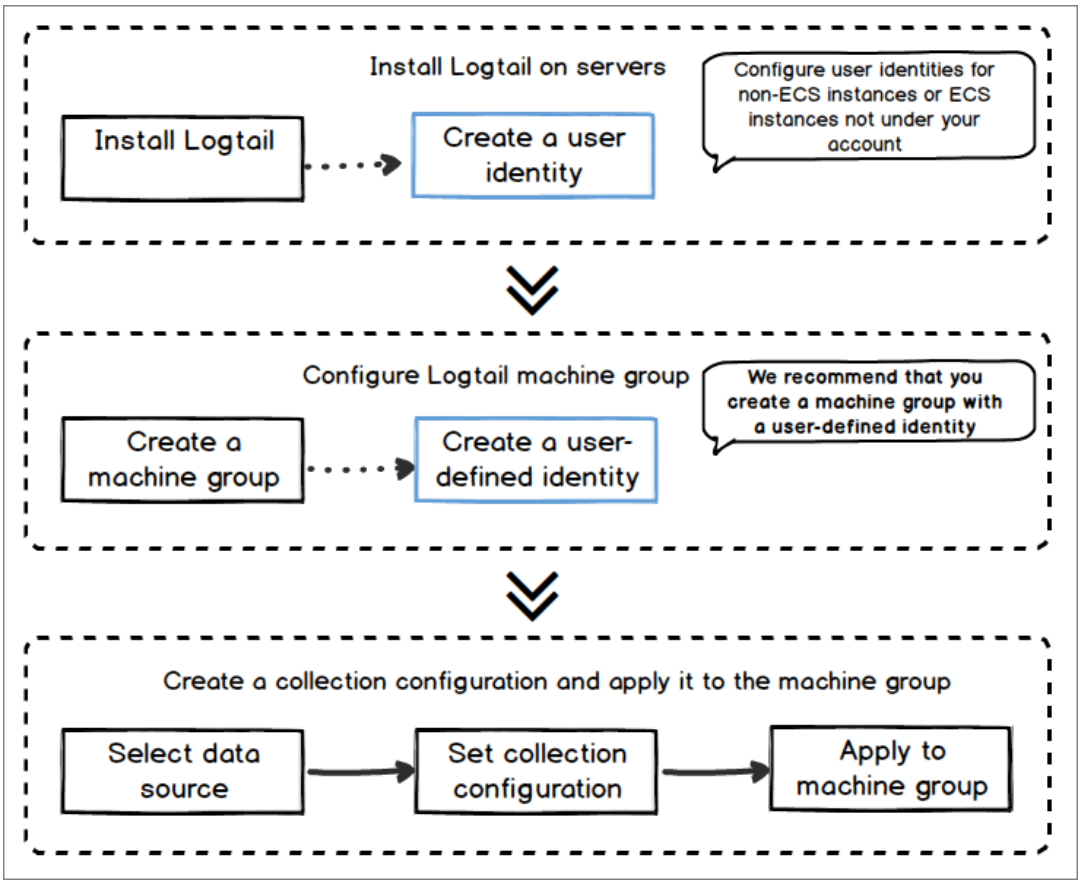
- Handles various exceptions that occur in the log collection process. If a network or server exception occurs, Logtail retries log collection and locally caches logs to ensure data security.
- Provides centralized management based on Log Service. After you install Logtail on the servers from which you want to collect logs, you can configure these servers and the collection method in the Log Service console. You do not need to log on to the servers.
- Provides a comprehensive self-protection mechanism. To minimize the impact of Logtail on the performance of the related servers, Logtail limits the usage of CPU, memory, and network resources.

Processing capabilities and limits

For more information, see Limits.

Configuration process

Configuration process



To collect logs from servers by using Logtail, follow these steps:

1. Install Logtail. For more information about how to install Logtail on a server from which you want to collect logs, see Install Logtail in Linux and Install Logtail in Windows.
2. Log Service use server groups to manage all servers from which you want to collect logs by using Logtail. Log Service allows you to define server groups by using IP addresses or custom identifiers. You can create a server group as prompted when you apply Logtail configurations to server groups.
3. Create a Logtail configuration and apply it to the server group. For more information about how to create a Logtail configuration, see Configure text log collection.

After the preceding process is complete, logs on the server are automatically collected and sent to the selected Logstore. However, historical logs are not collected. You can use the Log Service console, SDKs, or APIs to query these logs. Log Service allows you to view the status of log collection and check whether errors occur.

For more information, see Collect logs by using Logtail.

Containers

- For information about Alibaba Cloud Container Service for Kubernetes or user-created Kubernetes clusters, see Collect Kubernetes logs.
- For information about other user-created Docker clusters, see Collect standard Docker logs.

Terms

- Server group: A server group contains one or more servers from which logs of a specific type are collected. You can apply Logtail configurations to a server group. This enables Log Service to collect logs from all servers in the server group. You can use the Log Service console to manage a sever group. For example, you can create, delete, add, or remove a server. Each server group can contain different versions of Windows servers or Linux servers.
- Logtail: Logtail is the agent that collects logs from the servers on which Logtail runs. For more information, see Install Logtail in Linux and Install Logtail in Windows. After you install Logtail on a server, you must create a Logtail configuration and apply it to the server group to which the server belongs.
  - In Linux, Logtail is installed in the `/usr/local/ilogtail` directory. Logtail initiates two separate processes whose names start with ilogtail. One is a collection process and the other is a daemon process. The program running log is `/usr/local/ilogtail/ilogtail.LOG`.
  - In Windows, Logtail is installed in the `C:\Program Files\Alibaba\Logtail` (for 32-bit systems) or `C:\Program Files (x86)\Alibaba\Logtail` (for 64-bit systems) directory. You can choose Administrative Tools > Services to view the two Windows services generated from Logtail. One is LogtailWorker (log collection process) and the other is LogtailDaemon. The program running log is `logtail_*.log` in the installation directory.
- Logtail configuration: A Logtail configuration is a set of policies that are used by Logtail to collect logs. You can specify Logtail parameters such as the data source and collection mode. This allows you to customize log collection policies for all servers in a server group. A Logtail configuration determines how to collect a type of logs from a server, parse the logs, and send them to a specified Logstore. You can create a Logtail configuration for a Logstore in the Log Service console. This enables the Logstore to receive logs that are collected by using this Logtail configuration.

Features

Logtail provides the following features:

Feature	Description
Real-time log collection	Logtail dynamically monitors log files and reads and parses incremental logs in real time. In most cases, logs are sent to Log Service within 3 seconds after they are generated. **Note** Logtail does not collect historical data. If Logtail reads a log later than 12 hours after the log was generated, Logtail drops the log.
Automatic log rotation	Some applications rotate log files based on the file size or date. In the rotation process, the original log files are renamed and empty log files are created. For example, files such as *app.LOG.1* and *app.LOG.2* are generated for the *app.LOG* file after log rotation. You can specify the file (for example, *app.LOG*) to which collected logs are written. Logtail monitors the log rotation process to ensure that no logs are lost.
Multiple data sources	Logtail can collect text logs, syslogs, HTTP logs, and MySQL binlogs.

Feature	Description
Automatic exception handling	If data transmission fails due to exceptions such as Log Service errors, network errors, or quota exhaustion, Logtail retries log collection based on the specific scenario. If the retry fails, Logtail writes the data to the local cache and resends the data after the exception no longer exists.
Flexible collection policy configuration	You can create a Logtail configuration to specify how logs are collected from an ECS instance. You can select log directories and files by using exact match or wildcard match based on actual scenarios. You can also customize the extraction method of collected logs and the names of extracted fields. Log Service allows you to extract logs by using regular expressions. The log data models of Log Service require that each log has a precise timestamp. Logtail provides custom log time formats. This allows you to extract the required timestamps from log data of different formats.
Automatic synchronization of Logtail configurations	After you create or update a Logtail configuration in the Log Service console, Logtail receives and validates the configuration within 3 minutes. No data loss occurs during the configuration update process.
Automatic upgrade	After you install Logtail on a server, Log Service manages the automatic upgrade of Logtail without manual intervention. No data loss occurs during the Logtail upgrade process.
Status monitoring	Logtail monitors its consumption of CPU and memory in real time. This prevents Logtail from excessively consuming your resources. If the resource consumption exceeds the limit, Logtail restarts to avoid affecting other services on the server. Logtail limits network traffic to avoid excessive bandwidth consumption.
Data transmission with a signature	Logtail obtains your Alibaba Cloud AccessKey pair and uses it to sign all log data packets before they are sent. This prevents data tampering during transmission. **Note** Logtail obtains your Alibaba Cloud AccessKey pair over HTTPS to ensure the security of your AccessKey pair.

Data collection reliability

During data collection, Logtail stores the collected checkpoints to the local directory on a regular basis. If an exception (for example, an unexpected server shutdown or a process crash) occurs, Logtail restarts and then collects data from the last recorded checkpoint to prevent data loss. Logtail runs based on the resource limits specified in the configuration file. If the usage of a resource exceeds the limit for more than 5 minutes, Logtail restarts. After the restart, duplicate data may be generated.

Logtail uses multiple internal mechanisms to improve log collection reliability. However, logs may be lost in the following conditions:

- Logtail is not running but logs are rotated multiple times.
- The log rotation rate is high, for example, one rotation per second.
- The log collection rate is lower than the log generation rate for a long period of time.

28.1.3.1.1.2. Log collection process of Logtail

This topic describes the process that Logtail uses to collect server logs. The process consists of the following steps: monitor files, read files, process logs, filter logs, aggregate logs, and send logs.

> **Note** After the Logtail configuration is applied to a machine group, unmodified logs on the servers in the machine group are considered as historical logs. Logtail does not collect historical logs in normal running mode. If you want to collect historical logs, see Import historical logs.

Monitor files

After you install Logtail on a server and create a Logtail configuration based on a data source, the Logtail configuration is delivered from Log Service to Logtail in real time. Then, Logtail starts to monitor files based on the Logtail configuration.

1. Logtail scans log directories and files that comply with the specified file naming rules based on the specified log path and maximum monitoring directory depth.

   Logtail registers event monitoring and periodical polling for the directory from which Logtail collect logs. In Linux, Inotify is used. In Windows, ReadDirectoryChangesW is used. This method ensures the timeliness and stability of log collection.

2. If the compliant log files in the specified directory are not modified after the configuration is applied, Logtail does not collect these files. If modification events are generated for log files, Logtail triggers the collection process and reads these files.

Read files

After Logtail detects that a log file has been updated, Logtail reads the log file.

- If Logtail reads the log file for the first time, it checks the file size.
  - If the file size is less than 1 MB, Logtail reads the file from the beginning.
  - If the file size is greater than 1 MB, Logtail reads the file from the last 1 MB of data.
- If Logtail has read the file before, Logtail reads the file from the last checkpoint.
- Logtail can read up to 512 KB of data at a time. Therefore, you must limit the log size to 512 KB.

> **Notice** If you have changed the system time on your server, you must manually restart Logtail. Otherwise, the log time is incorrect and logs are dropped.

Process logs

When Logtail reads a log, it divides the log into multiple lines, parses the log, and sets the time field of the log.

- Divide each log into multiple lines

  If you specifies **Regex to Match First Line Only** in the Logtail configuration, the log data read by Logtail at one time is divided into multiple lines based on the specified beginning of the line. If the beginning of the line is not specified, each data block is processed as a log.

- Parse each log

  Logtail parses each log based on the Logtail configuration, such as regular expressions, delimiters, and JSON.

  > **Notice** A complicated regular expression may lead to high CPU usage. Therefore, we recommend that you use an efficient regular expression.

- Handle parsing failures

  Logtail determines how to handle parsing failures based on whether the Drop Failed to Parse Logs switch is turned on in the Logtail configuration.

  - If the **Drop Failed to Parse Logs** is turned on, Logtail drops the logs that fail to be parsed and reports an error.

  - If the **feature** switch is turned off, Logtail uploads the logs that fail to be parsed. In these logs, the key is set to **raw_log** and the value is set to the log content.

- Set the time field of a log
  - If the time field of the log is not specified, the log time is the current parsing time.
  - If the time field of the log is specified, the following operations are performed:
    - The log time is extracted from the parsed log fields if the difference between the log time and the current time is less than 12 hours.
    - The log is dropped and an error is reported if the difference between the log time and the current time is greater than 12 hours.

Filter logs

After Logtail process logs, it filters the logs based on the filter configuration.

- If the **Filter Configuration** is not specified, Logtail does not filter logs and proceed with the next step.
- If the **Filter Configuration** is specified, Logtail traverses and verifies all the fields of each log.
  - Logtail collects a log if the log matches the filter configuration.
  - Logtail does not collect a log if the log does not match the filter configuration.

Aggregate logs

Logtail sends the logs that match the filter configuration to Log Service. To reduce the number of network requests, Logtail caches the processed and filtered logs for a period of time. Then, Logtail aggregates and packages these logs before sending them to Log Service.

If one of the following conditions is met during caching, logs are immediately packaged and sent to Log Service.

- The log aggregation period exceeds three seconds.
- The number of aggregated logs exceeds 4,096.
- The total size of aggregated logs exceeds 512 KB.

Send logs

Logtail sends the aggregated logs to Log Service. You can set the `max_bytes_per_sec` and `send_request_concurrency` parameters in Set Logtail startup parameters to adjust the log data sending rate and the maximum concurrent requests. In this case, Logtail ensures that the sending rate and the concurrent requests do not exceed the limits.

If the log data fails to be sent, Logtail retries or stops the operation based on the error message.

Error	Description	Handling method
401	Logtail is not authorized to collect data.	Logtail drops the log packets.
404	The specified project or Logstore does not exist in the Logtail configuration.	Logtail drops the log packets.
403	The shard quota is exhausted.	Logtail waits for three seconds and retries.
500	A Log Service exception has occurred.	Logtail waits for three seconds and retries.
Network timeout	A network connection error has occurred.	Logtail waits for three seconds and retries.

28.1.3.1.1.3. Logtail configuration files and record files

This topic describes the basic configuration files and record files of Logtail. When Logtail is active, it uses a series of configuration files and generates record files.

The basic configuration files are as follows:

- Startup configuration file (ilogtail_config.json)
- Account ID configuration file
- User-defined identifier file (user_defined_id)
- Logtail configuration file (user_log_config.json)

The basic record files are as follows:

- AppInfo record file (app_info.json)
- Logtail operational log file (ilogtail.LOG)
- Logtail plug-in log file (logtail_plugin.LOG)
- Container path mapping file (docker_path_config.json)

Startup configuration file (ilogtail_config.json)

This file is used to query or set Logtail runtime parameters. The file is in the JSON format.

After you install Logtail, you can use the startup configuration file to perform the following operations:

- Change the values of the Logtail runtime parameters.

  You can change the CPU usage threshold, usage threshold of terminate and stay resident (TSR) programs, and other settings.

- Check whether the installation commands are correct.

  The settings of `config_server_address` and `data_server_list` in this file depend on the parameters and installation commands selected when you installed Logtail. If the region specified in config_server_address is unreachable or is different from the region where Log Service resides, the selected parameters or commands are incorrect. In this case, Logtail cannot collect logs and must be reinstalled.

> **Warning**
> - The file must be a valid JSON file. Otherwise, Logtail cannot be started.
> - If you modify the file, you must restart Logtail to validate your modifications.

The following table describes the default parameters in the startup configuration file. You can also add other parameters. For more information, see Set Logtail startup parameters.

Default parameters

Parameter	Description
config_server_address	The address that Logtail uses to receive the configuration file from Log Service. This address depends on the parameters and installation commands that you selected when you installed Logtail. Ensure that the address is reachable and is in the same region as Log Service.
data_server_list	The data server address. This address depends on the parameters and installation commands that you selected when you installed Logtail. Ensure that the address is reachable and is in the same region as Log Service.
cluster	The name of the region where a server resides.
endpoint	The endpoint of Log Service. For more information, see View the information of a project.
cpu_usage_limit	The CPU usage threshold, which is calculated by core.

Parameter	Description
mem_usage_limit	The TSR usage threshold.
max_bytes_per_sec	The traffic limit on the raw data that is sent by Logtail. If the value of this parameter is greater than 20 Mbit/s, traffic limiting does not take effect.
process_thread_count	The number of threads that Logtail uses to write data to log files.
send_request_concurrency	The number of concurrent requests for sending data packets asynchronously. Logtail sends data packets asynchronously by default. If the write transactions per second (TPS) is high, you can set a greater value for this parameter.

- File path
  - Linux: The file is stored in */usr/local/ilogtail/ilogtail_config.json*.
  - Container Service: The file is stored in a Logtail container. The file path is specified in the environment variable `ALIYUN_LOGTAIL_CONFIG` of the Logtail container. You can run the command `docker inspect ${logtail_container_name} | grep ALIYUN_LOGTAIL_CONFIG` to query the file path. For example, the file path is */etc/ilogtail/conf/cn-hangzhou/ilogtail_config.json*.
  - Windows:
    - 64-bit: The file is stored in *C:\Program Files (x86)\Alibaba\Logtail\ilogtail_config.json*.
    - 32-bit: The file is stored in *C:\Program Files\Alibaba\Logtail\ilogtail_config.json*.
- Sample file

```
$cat /usr/local/ilogtail/ilogtail_config.json
{
    "config_server_address" : "http://logtail.cn-hangzhou-intranet.log.aliyuncs.com",
    "data_server_list" :
    [
        {
            "cluster" : "ap-southeast-2",
            "endpoint" : "cn-hangzhou-intranet.log.aliyuncs.com"
        }
    ],
    "cpu_usage_limit" : 0.4,
    "mem_usage_limit" : 100,
    "max_bytes_per_sec" : 2097152,
    "process_thread_count" : 1,
    "send_request_concurrency" : 4,
    "streamlog_open" : false
}
```

Account ID configuration file

This file contains the ID of your Apsara Stack tenant account. The file indicates that the account can collect logs from the server where Logtail is installed. If you want to collect logs from ECS instances that do not belong to your account or from on-premises data centers, you must create an account ID configuration file.

> **Note**
> - The file is used only when you collect logs from ECS instances that do not belong to your account and or from on-premises data centers.
> - The file can contain only the ID of your Apsara Stack tenant account. It cannot contain the IDs of RAM users under your Apsara Stack tenant account.
> - The file name cannot contain a suffix.
> - Each Logtail can have multiple account ID configuration files. Each Logtail container can have only one account ID configuration file.

- File path
  - Linux: The file is stored in `/etc/ilogtail/users/`.
  - Container Service: The file is stored in a Logtail container. The file path is specified in the environment variable `ALIYUN_LOGTAIL_USER_ID` of the Logtail container. You can run the command `docker inspect ${logtail_container_name} | grep ALIYUN_LOGTAIL_USER_ID` to query the file path.
  - Windows: The file is stored in `C:\LogtailData\users\`.
- Sample file

```
$ls /etc/ilogtail/users/
*************** ****************
```

User-defined identifier file (user_defined_id)

This file is used to configure machine groups with user-defined identifiers. For more information, see Create a machine group based on a custom ID.

> **Note**
> - This file is used only when you configure a machine group with user-defined identifiers.
> - If you configure multiple user-defined identifiers for a machine group, separate them with line breaks.

- File path
  - Linux: The file is stored in */etc/ilogtail/user_defined_id*.
  - Container Service: The file is stored in a Logtail container. The file path is specified in the environment variable `ALIYUN_LOGTAIL_USER_DEFINED_ID` of the Logtail container. You can run the command *docker inspect ${logtail_container_name} | grep ALIYUN_LOGTAIL_USER_DEFINED_ID* to query the file path.
  - Windows: The file is stored in *C:\LogtailData\user_defined_id*.
- Sample file

```
$cat /etc/ilogtail/user_defined_id
aliyun-ecs-rs1e16355
```

Logtail configuration file (user_log_config.json)

This file contains the Logtail configuration that Logtail receives from Log Service. The file is in the JSON format and is updated along with configuration updates. You can use this file to check whether the Logtail configuration is delivered to the server where Logtail is installed. If the Logtail configuration file exists and all contents in the file are up to date, the Logtail configuration is delivered.

> **Notice**
>
> - We recommend that you do not modify the Logtail configuration file unless you need to specify sensitive information, such as the AccessKey pair and database password.
> - You must upload this file when you submit a ticket.

- File path
  - Linux: The file is stored in */usr/local/ilogtail/user_log_config.json*.
  - Container Service: The file is stored in */usr/local/ilogtail/user_log_config.json*.
  - Windows
    - 64-bit: The file is stored in *C:\Program Files (x86)\Alibaba\Logtail\user_log_config.json*.
    - 32-bit: The file is stored in *C:\Program Files\Alibaba\Logtail\user_log_config.json*.
- Sample file

```
$cat /usr/local/ilogtail/user_log_config.json
{
   "metrics" : {
      "##1.0##k8s-log-c12ba2028*****939f0b$app-java" : {
         "aliuid" : "16542189*****50",
         "category" : "app-java",
         "create_time" : 1534739165,
         "defaultEndpoint" : "cn-hangzhou-intranet.log.aliyuncs.com",
         "delay_alarm_bytes" : 0,
         "enable" : true,
         "enable_tag" : true,
         "filter_keys" : [],
         "filter_regs" : [],
         "group_topic" : "",
         "local_storage" : true,
         "log_type" : "plugin",
         "log_tz" : "",
         "max_send_rate" : -1,
         "merge_type" : "topic",
         "plugin" : {
            "inputs" : [
               {
                  "detail" : {
                     "IncludeEnv" : {
                        "aliyun_logs_app-java" : "stdout"
                     },
                     "IncludeLable" : {
                        "io.kubernetes.container.name" : "java-log-demo-2",
                        "io.kubernetes.pod.namespace" : "default"
                     },
                     "Stderr" : true,
                     "Stdout" : true
                  },
                  "type" : "service_docker_stdout"
               }
            ]
         },
         "priority" : 0,
         "project_name" : "k8s-log-c12ba2028c*****ac1286939f0b",
         "raw_log" : false,
         "region" : "cn-hangzhou",
         "send_rate_expire" : 0,
         "sensitive_keys" : [],
         "tz_adjust" : false,
         "version" : 1
      }
   }
}
```

AppInfo record file (app_info.json)

This file contains the startup time of Logtail. It also contains the IP address and hostname that Logtail obtains. You must check the IP address obtained by Logtail when you configure an IP address-based machine group.

In most cases, Logtail obtains server IP addresses based on the following rules:

- If the IP address of a server is associated with its hostname in the */etc/hosts* server file, Logtail obtains the IP address.
- If the IP address of a server is not associated with its hostname, Logtail obtains the IP address of the first

network interface card (NIC) on the server.

> **Note**
> - The AppInfo record file contains only the basic Logtail information, which cannot be manually modified.
> - If you modify the hostname or other network settings of the server, you must restart Logtail to obtain a new IP address.

Parameters

Parameter	Description
UUID	The serial number of the server.
hostname	The hostname.
instance_id	The unique identifier of Logtail. This identifier is randomly generated.
ip	The IP address that is obtained by Logtail. If this parameter is not specified, Logtail has not obtained the IP address of a server. In this case, Logtail cannot function properly. You must set an IP address for your server and restart Logtail. **Note** If the machine group is an IP address-based machine group, ensure that the IP address specified for the machine group is the same as the value of this parameter. If the two IP address are different, modify the IP address that you specified for the machine group in the Log Service console. Check the IP addresses again after 1 minute.
logtail_version	The version of Logtail.
os	The version of the operating system.
update_time	The last startup time of Logtail.

- File path
  - Linux: The file is stored in */usr/local/ilogtail/app_info.json*.
  - Container Service: The file is stored in */usr/local/ilogtail/app_info.json*.
  - Windows
    - 64-bit: The file is stored in *C:\Program Files (x86)\Alibaba\Logtail\app_info.json*.
    - 32-bit: The file is stored in *C:\Program Files\Alibaba\Logtail\app_info.json*.
- Sample file

```
$cat /usr/local/ilogtail/app_info.json
{
   "UUID" : "",
   "hostname" : "logtail-ds-slpn8",
   "instance_id" : "E5F93BC6-B024-11E8-8831-0A58AC14039E_1**. ***. ***.***_1536053315",
   "ip" : "1**. ***. ***.***",
   "logtail_version" : "0.16.13",
   "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
   "update_time" : "2018-09-04 09:28:36"
}
```

Logtail operational log file (ilogtail.LOG)

This file contains operational information about Logtail. Log severity levels are ranked as follows in ascending order: `INFO`, `WARN`, `ERROR`. Logs of the `INFO` level can be ignored.

- File path
  - For Linux: The file is stored in */usr/local/ilogtail/ilogtail.LOG*.
  - Container Service: The file is stored in */usr/local/ilogtail/ilogtail.LOG*.
  - Windows
    - 64-bit: The file is stored in *C:\Program Files (x86)\Alibaba\Logtail\logtail_*.log*.
    - 32-bit: The file is stored in *C:\Program Files\Alibaba\Logtail\logtail_*.log*.
- Sample file

```
$tail /usr/local/ilogtail/ilogtail.LOG
[2018-09-13 01:13:59.024679]    [INFO]    [3155]    [build/release64/sls/ilogtail/elogtail.cpp:123
]    change working dir:/usr/local/ilogtail/
[2018-09-13 01:13:59.025443]    [INFO]    [3155]    [build/release64/sls/ilogtail/AppConfig.cpp:17
5]    load logtail config file, path:/etc/ilogtail/conf/ap-southeast-2/ilogtail_config.json
[2018-09-13 01:13:59.025460]    [INFO]    [3155]    [build/release64/sls/ilogtail/AppConfig.cpp:17
6]    load logtail config file, detail:{
   "config_server_address" : "http://logtail.ap-southeast-2-intranet.log.aliyuncs.com",
   "data_server_list" : [
      {
         "cluster" : "ap-southeast-2",
         "endpoint" : "ap-southeast-2-intranet.log.aliyuncs.com"
      }
]
```

Logtail plug-in log file (logtail_plugin.LOG)

This file contains operational information about plug-ins, such as stdout, binlog, and HTTP plug-ins. Log severity levels are ranked as follows in ascending order: `INFO`, `WARN`, `ERROR`. Logs of the `INFO` level can be ignored.

- File path
  - Linux: The file is stored in */usr/local/ilogtail/logtail_plugin.LOG*
  - Container Service: The file is stored in */usr/local/ilogtail/logtail_plugin.LOG*.
  - Windows: The file is not supported.
- Sample file

```
$tail /usr/local/ilogtail/logtail_plugin.LOG
2018-09-13 02:55:30 [INF] [docker_center.go:525] [func1] docker fetch all:start
2018-09-13 02:55:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop
2018-09-13 03:00:30 [INF] [docker_center.go:525] [func1] docker fetch all:start
2018-09-13 03:00:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop
2018-09-13 03:03:26 [INF] [log_file_reader.go:221] [ReadOpen] [##1.0##sls-zc-test-hz-pub$docker-stdout-config,k8s-stdout]    open file for read, file:/logtail_host/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log    offset:40379573    status:794354-64769-40379963
2018-09-13 03:03:26 [INF] [log_file_reader.go:221] [ReadOpen] [##1.0##k8s-log-c12ba2028cfb444238cd9ac1286939f0b$docker-stdout-config,k8s-stdout]    open file for read, file:/logtail_host/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log    offset:40379573    status:794354-64769-40379963
2018-09-13 03:04:26 [INF] [log_file_reader.go:308] [CloseFile] [##1.0##sls-zc-test-hz-pub$docker-stdout-config,k8s-stdout]    close file, reason:no read timeout    file:/logtail_host/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log    offset:40379963    status:794354-64769-40379963
2018-09-13 03:04:27 [INF] [log_file_reader.go:308] [CloseFile] [##1.0##k8s-log-c12ba2028cfb444238cd9ac1286939f0b$docker-stdout-config,k8s-stdout]    close file, reason:no read timeout    file:/logtail_host/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log    offset:40379963    status:794354-64769-40379963
2018-09-13 03:05:30 [INF] [docker_center.go:525] [func1] docker fetch all:start
2018-09-13 03:05:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop
```

Container path mapping file (docker_path_config.json)

This file is automatically created only when container files are collected. It records path mappings between container files and actual files. The file is in the JSON format.

> **Note** This file is only an information record file. Modifications to this file do not take effect. If you delete this file, another one is automatically created without service interruptions.

- File path

  The file is stored in */usr/local/ilogtail/docker_path_config.json*.

- Sample file

```
$cat /usr/local/ilogtail/docker_path_config.json
{
   "detail" : [
      {
         "config_name" : "##1.0##k8s-log-c12ba2028cfb444238cd9ac1286939f0b$nginx",
         "container_id" : "df19c06e854a0725ea7fca7e0378b0450f7bd3122f94fe3e754d8483fd330d10",
         "params" : "{\n   \"ID\" : \"df19c06e854a0725ea7fca7e0378b0450f7bd3122f94fe3e754d8483fd330d10\",\n   \"Path\" : \"/logtail_host/var/lib/docker/overlay2/947db346695a1f65e63e582ecfd10ae1f57019a1b99260b6c83d00fcd1892874/diff/var/log\",\n   \"Tags\" : [\n      \"nginx-type\",\n      \"access-log\",\n      \"_image_name_\",\n      \"registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest\",\n      \"_container_name_\",\n      \"nginx-log-demo\",\n      \"_pod_name_\",\n      \"nginx-log-demo-h2lzc\",\n      \"_namespace_\",\n      \"default\",\n      \"_pod_uid_\",\n      \"87e56ac3-b65b-11e8-b172-00163f008685\",\n      \"_container_ip_\",\n      \"172.20.4.224\",\n      \"purpose\",\n      \"test\"\n   ]\n}\n"
      }
   ],
   "version" : "0.1.0"
}
```

28.1.3.1.2. Installation

28.1.3.1.2.1. Install Logtail in Linux

This topic describes how to install Logtail on a Linux server.

Supported systems

Logtail supports the following x86-64 (64-bit) Linux operating systems:

- Aliyun Linux
- Ubuntu
- Debian
- CentOS
- openSUSE
- Red Hat

Procedure

> **Note** If you have installed Logtail, the installer will uninstall the existing version of Logtail, delete the `/usr/local/ilogtail` directory, and then reinstall Logtail. By default, Logtail runs after the installation and at startup.

1. Run the following command to download the Logtail installer:

   ```
   wget http://${service:sls-backend-server:sls_data.endpoint}/logtail.sh -O logtail.sh; chmod 755 logtail.sh
   ```

   > **Note** You must replace `${service:sls-backend-server:sls_data.endpoint}` in the command with the actual endpoint. For more information about endpoints, see View the information of a project.

2. Run the installation command.

   Start Linux PowerShell and run the following command as an administrator to install Logtail:

   ```
   ./logtail.sh install
   ```

3. Configure an account ID for a server.

View the version of Logtail

To view the version of Logtail, open the file in the `/usr/local/ilogtail/app_info.json` directory. The `logtail_version` field shows the version of Logtail.

```
$cat /usr/local/ilogtail/app_info.json
{
   "UUID" : "0DF18E97-0F2D-486F-B77F-*********",
   "hostname" : "david*******",
   "instance_id" : "F4FAFADA-F1D7-11E7-846C-00163E30349E_*********_1515129548",
   "ip" : "**********",
   "logtail_version" : "0.16.0",
   "os" : "Linux; 2.6.32-220.23.2.ali1113.el5.x86_64; #1 SMP Thu Jul 4 20:09:15 CST 2013; x86_64",
   "update_time" : "2018-01-05 13:19:08"
}
```

Upgrade Logtail

You can use the Logtail installer (logtail.sh) to upgrade Logtail. The installer selects an upgrade method based on the configurations of the existing Logtail.

> **Note** During the upgrade, Logtail is temporarily stopped. Only related files are overwritten. The configuration file, checkpoint file, and logs are retained.

Run the following commands to upgrade Logtail:

```
# Download the Logtail installer.
wget http://${service:sls-backend-server:sls_data.endpoint}/logtail.sh -O logtail.sh; chmod 755 logtail.sh
# Upgrade Logtail.
sudo ./logtail.sh upgrade
```

Response:

```
# The upgrade is successful.
Stop logtail successfully.
ilogtail is running
Upgrade logtail success
{
   "UUID" : "***",
   "hostname" : "***",
   "instance_id" : "***",
   "ip" : "***",
   "logtail_version" : "0.16.11",
   "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
   "update_time" : "2018-08-29 15:01:36"
}
# The upgrade fails because the current version is the latest version.
[Error]:    Already up to date.
```

Start and stop Logtail

- Start Logtail

  Run the following command as an administrator:

  ```
  /etc/init.d/ilogtaild start
  ```

- Stop Logtail

  Run the following command as an administrator:

  ```
  /etc/init.d/ilogtaild stop
  ```

Uninstall Logtail

Run Linux PowerShell as an administrator to uninstall Logtail:

```
wget http://${service:sls-backend-server:sls_data.endpoint}/logtail.sh -O logtail.sh
chmod 755 logtail.sh
./logtail.sh uninstall
```

28.1.3.1.2.2. Install Logtail in Windows

This topic describes how to install Logtail on a Windows server.

Supported systems

Logtail supports the following Windows operating systems:

- Windows 7 (Client) 32-bit
- Windows 7 (Client) 64-bit
- Windows Server 2008 32-bit
- Windows Server 2008 64-bit
- Windows Server 2012 64-bit
- Windows Server 2016 64-bit

Procedure

1. Download the installation package.

   Run the following command to download the installation package:

   ```
   wget http://${service:sls-backend-server:sls_data.endpoint}/windows/logtail_installer.zip
   ```

   > **Note** You must replace `${service:sls-backend-server:sls_data.endpoint}` in the command with the actual endpoint. For more information about endpoints, see View the information of a project.

2. Decompress the `logtail_installer.zip` package to the current directory.
3. Run the installation command.

   Run Windows PowerShell or Command Prompt as an administrator. Enter the `logtail_installer` directory, and then run the installation command based on the network type.

   ```
   .\logtail_installer.exe install me-east-1
   ```

   > **Note** You must replace `${region}` in the command with the actual endpoint. For more information about endpoints, see View the information of a project.

4. Configure an account ID for a server.

Installation directory

After you run the installation command, Logtail is installed in the specified directory. The directory cannot be changed. In the directory, you can View the version of Logtail in the *app_info.json* file or Uninstall Logtail.

The installation directory is as follows:

- 32-bit Windows: *C:\Program Files\Alibaba\Logtail*
- 64-bit Windows: *C:\Program Files (x86)\Alibaba\Logtail*

> **Note** You can run 32-bit or 64-bit applications in a 64-bit Windows operating system. However, the operating system stores 32-bit applications in separate x86 folders to ensure compatibility.
>
> Logtail for Windows is a 32-bit application. Therefore, it is installed in the *Program Files (x86)* folder in 64-bit Windows. If Logtail for 64-bit Windows becomes available in the future, it will be installed in the *Program Files* folder.

View the version of Logtail

To view the version of Logtail, go to the default installation directory, and then use the notepad or another text editor to open the *app_info.json* file. The `logtail_version` field shows the version of Logtail.

In the following example, the version of Logtail is 1.0.0.0:

```
{
    "logtail_version" : "1.0.0.0"
}
```

Upgrade Logtail

- Automatic upgrade

  Logtail later than 1.0.0.0 is automatically upgraded in Windows.

- Manual upgrade

  Logtail earlier than 1.0.0.0 must be manually upgraded. The manual upgrade procedure is the same as the installation procedure.

  > **Note** During a manual upgrade, the files in the original installation directory are deleted. We recommend that you back up the files before you perform a manual upgrade.

Start and stop Logtail

Open the **Control Panel**, choose System and Security > **Administrative Tools**, and then double-click **Services**.

Find the service based on your Logtail version.

- Logtail 0.x.x.x: LogtailWorker.
- Logtail 1.0.0.0 and later: LogtailDaemon.

Perform the following operations as required:

- Start Logtail

  Right-click the service and select **Start** from the shortcut menu.

- Stop Logtail

  Right-click the service and select **Stop** from the shortcut menu.

- Restart Logtail

  Right-click the service and select **Restart** from the shortcut menu.

Uninstall Logtail

Run Windows PowerShell or Command Prompt as an administrator. Enter the `logtail_installer` directory, and then run the following command:

```
.\logtail_installer.exe uninstall
```

After Logtail is uninstalled, the installation directory is deleted. However, some residual configuration data is still maintained in the *C:\LogtailData* directory. You can manually delete the data. The residual configuration data includes the following information:

- *checkpoint*: checkpoints of all plug-ins, for example, the Windows event log plug-in.
- *logtail_check_point*: checkpoints of Logtail.
- *users*: IDs of Apsara Stack tenant accounts.

28.1.3.1.2.3. Set Logtail startup parameters

This topic describes how to set Logtail startup parameters.

Context

You may need to set Logtail startup parameters in the following scenarios:

- You need to collect a large number of log files that consume much memory. You want to maintain the metadata (such as the file signature, collection location, and file name) of each file in the memory.
- The CPU usage is high due to heavy log data traffic.
- The traffic sent to Log Service is heavy due to a large amount of log data.
- You want to collect syslogs or TCP data streams.

Startup configurations

- File path

```
/usr/local/ilogtail/ilogtail_config.json
```

- File format

  JSON

- Sample file (only partial configurations are provided)

```
{
    ...
    "cpu_usage_limit" : 0.4,
    "mem_usage_limit" : 100,
    "max_bytes_per_sec" : 2097152,
    "process_thread_count" : 1,
    "send_request_concurrency" : 4,
    "streamlog_open" : false,
    "streamlog_pool_size_in_mb" : 50,
    "streamlog_rcv_size_each_call" : 1024,
    "streamlog_formats":[],
    "streamlog_tcp_port" : 11111,
    "buffer_file_num" : 25,
    "buffer_file_size" : 20971520,
    "buffer_file_path" : "",
    ...
}
```

Startup parameters

Parameter	Description	Example
cpu_usage_limit	The CPU usage threshold for a single core. Data type: double.	For example, the value 0.4 indicates that the CPU usage of Logtail is limited to 40% processing capacity of a single core. In most cases, the processing capacity of a single core is about 24 MB/s in the simple mode and 12 MB/s in the full regex mode.
mem_usage_limit	The usage threshold of the resident memory. Data type: integer. Unit: MB.	For example, the value 100 indicates that the memory usage of Logtail is limited to 100 MB. If the threshold is exceeded, Logtail restarts. If you want to collect more than 1,000 log files, you can increase the threshold value.
max_bytes_per_sec	The traffic limit on the raw data that is sent by Logtail. Data type: integer. Unit: bytes/s.	For example, the value 2097152 indicates that the data transfer rate of Logtail is limited to 2 MB/s.

Parameter	Description	Example
process_thread_count	The number of threads that Logtail uses to process data.	Default value: 1. Each thread provides a write speed of 24 MB/s in the simple mode and 12 MB/s in the full regex mode. We recommend that you do not modify the default value.
send_request_concurrency	Logtail sends data packets asynchronously by default. If the write transactions per second (TPS) is high, you can set this parameter to a greater value.	Twenty asynchronous concurrencies are provided by default. Each concurrency can provide 0.5 MB/s to 1 MB/s network throughput. The number of concurrencies varies with the network delay.
streamlog_open	Specifies whether to receive syslogs. Data type: Boolean.	The value false indicates that syslogs are not received. The value true indicates that syslogs are received.
streamlog_pool_size_in_mb	The size of memory pool that the syslog server uses to cache syslogs. Unit: MB.	Logtail requests memory when it starts. Set the memory pool size based on the server memory size and your business requirements.
streamlog_rcv_size_each_call	The size of the buffer that Logtail uses when the linux socket rcv API is called. Unit: bytes. Valid values: 1024 to 8192.	You can set a greater value if the syslog traffic is high.
streamlog_formats	The method that is used to parse received syslogs.	N/A
streamlog_tcp_addr	The associated address that Logtail uses to receive syslogs. Default value: 0.0.0.0.	N/A
streamlog_tcp_port	The TCP port that Logtail uses to receive syslogs.	Default value: 11111.
buffer_file_num	The maximum number of cached files. If a network exception occurs or the writing quota is exceeded, Logtail writes parsed logs to local files in the installation directory. After the network recovers or a new writing quota is available, Logtail retries to send the logs to Log Service.	Default value: 25.
buffer_file_size	The maximum number of bytes that can be contained in each cache file. The maximum disk space available for cache files is the value of buffer_file_num multiplied by the value of buffer_file_size.	Default value: 20971520 bytes (20 MB).

Parameter	Description	Example
buffer_file_path	The directory in which cached files are stored. If you modify this parameter, you must move the files (for example, *logtail_buffer_file_**) in the old cache directory to the new directory. Then, Logtail can read, send, and delete the cache files.	The default value is null, which indicates that the cached files are stored in the Logtail installation directory */usr/local/ilogtail*.
bind_interface	The name of the NIC associated with the local machine, for example, eth1. This parameter is valid only for Logtail that runs in Linux.	By default, the available NICs are automatically associated with the local machine. If you specify this parameter, Logtail will use the specified NIC to upload logs.
check_point_filename	The full path in which the checkpoint file is stored. This parameter is used to customize the path to store the checkpoint file of Logtail.	Default value: */tmp/logtail_check_point*. We recommend that Docker users modify this path and mount the directory where the checkpoint file resides to the host. Otherwise, duplicate collection occurs due to checkpoint data loss when the container is released. For example, you can set check_point_filename to `/data/logtail/check_point.dat` in Docker and add `-v /data/docker1/logtail:/data/logtail` to the Docker startup command. Then, the */data/docker1/logtail* directory of the host is mounted to the */data/logtail* directory of Docker.

Note

- The preceding table lists only the common startup parameters. If the *ilogtail_config.json* file contains parameters that are not listed in the table, the default settings are used for these parameters.
- We recommend that you do not add unnecessary parameters to the *ilogtail_config.json* file.

Modify configurations

1. Configure the *ilogtail_config.json* file as needed.

   Ensure that the modified configurations are in the valid JSON format.

2. Restart Logtail to apply the modified configurations.

```
/etc/init.d/ilogtaild stop
/etc/init.d/ilogtaild start
/etc/init.d/ilogtaild status
```

28.1.3.1.3. Logtail machine group

28.1.3.1.3.1. Overview

Log Service uses machine groups to manage the servers from which you want to collect logs by using Logtail.

A machine group is a virtual group that contains multiple servers. If you want to use a Logtail configuration file to collect logs from multiple servers, you can add the servers to a machine group. Then, you can apply the Logtail configuration file to the machine group.

To define a machine group, you can use one of the following methods:

- IP address: Add the IP addresses of all servers to a machine group. Each server can be identified by using its unique IP address.
- Custom ID: Use a custom ID to identify the machine group and use the same ID for servers in the machine group.

> **Note** Windows and Linux servers cannot be added to the same machine group.

IP address-based machine groups

You can add multiple servers to a machine group by adding their IP addresses to the machine group. Then, you can create a Logtail configuration file for all the servers at the same time.

- If you use ECS instances and have not associated them with hostnames or changed their network types, you can add their private IP addresses to the machine group.
- In other cases, you must add the server IP addresses obtained by Logtail to a machine group. The IP address of each server is recorded in the IP address field of the *app_info.json* file on the server.

> **Note** The *app_info.json* file records the internal information of Logtail. This file includes the server IP addresses obtained by Logtail. If you modify the IP address field of the file, the IP addresses obtained by Logtail remain unchanged.

Logtail obtains a server IP address by using the following methods:

- If the IP address of a server is associated with the hostname in the */etc/hosts* file of the server, Logtail obtains this IP address.
- If the IP address of a server is not associated with the hostname, Logtail obtains the IP address of the first network interface controller (NIC) on the server.

For more information, see Create an IP address-based server group.

Custom ID-based machine groups

You can use custom IDs to dynamically define machine groups.

An application system consists of multiple modules. You can scale out each module by adding multiple servers to the module. If you want to collect logs by module, you can create a machine group for each module. Therefore, you must specify a custom ID for each server in each module. For example, a website consists of an HTTP request processing module, a caching module, a logic processing module, and a storage module. The custom IDs of these modules can be http_module, cache_module, logic_module, and store_module.

For more information, see Create a machine group based on a custom ID.

28.1.3.1.3.2. Create a machine group based on a server IP address

This topic describes how to create a machine group based on a server IP address. You can create a machine group based on a server IP address that is obtained by using a Logtail configuration file. You can then use the same Logtail configuration file to collect logs from the machine group.

Prerequisites

- A project is created. A Logstore is created in the project.
- One or more servers are available. The IP addresses of the servers are obtained.
- Logtail is installed on the servers. For more information, see Install Logtail in Linux or Install Logtail in Windows.
- The ID of your Apsara Stack tenant account is configured on the server. For more information, see Configure an account ID for a server.

Procedure

1. Log on to the Log Service console.
2. In the **Projects** section, click the name of a project.
3. In the left-side navigation pane, click the **Machine Groups** icon.
4. In the pane that appears, click the icon next to **Machine Group** and select **Create Machine Group** from the shortcut menu.

   You can also create a machine group in the Logtail configuration wizard.
5. Create a machine group.
   i. Enter a machine group name in the **name** field.

   The machine group name must be 3 to 128 characters in length and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit.

   > **Notice** After the machine group is created, its name cannot be modified.

   ii. Select **IP Addresses** from the **Identifier** drop-down list.
   iii. Enter a topic name in the **Topic** field.

   For more information about topics, see Generate a topic.
   iv. Enter the IP addresses of the servers in the **IP Addresses** text box.

   > **Notice**
   > - Separate each IP address with a line break.
   > - Do not add Windows servers and Linux servers to the same machine group.
6. Click **OK**.

Result

You can view the created machine group in the **Machine Groups** pane.

28.1.3.1.3.3. Create a machine group based on a custom ID

This topic describes how to create a machine group based on a custom ID.

Context

You can use a custom ID to identify a machine group in the following scenarios:

- Servers reside in multiple custom network environments such as virtual private clouds (VPCs). IP addresses of different servers may be the same. In this scenario, Log Service cannot distinguish between servers based on IP addresses.
- You want to implement automatic server discovery. To do this, you only need to set a custom ID of a new server to the custom ID of an existing machine group. Log Service automatically identifies the server and adds it to the machine group.

Procedure

1. Set a custom ID on a server.
   - **Linux Logtail**

     Set a custom ID in the `/etc/ilogtail/user_defined_id` file.

     For example, if you need to set a custom ID of a server to `userdefined`, run the following command to open the file:

     ```
     # vim /etc/ilogtail/user_defined_id
     ```

     In the file, enter `userdefined`.
   - **Windows Logtail**

     Set a custom ID in the `C:\LogtailData\user_defined_id` file.

     For example, if you need to set a custom ID of a server, run the following command:

     ```
     C:\LogtailData>more user_defined_id
     userdefined_windows
     ```

   > **Notice**
   > - A machine group cannot include both Linux and Windows servers. Therefore, do not set a custom ID of a Linux server and a custom ID of a Windows server to the same value.
   > - You can set multiple custom IDs for a single server. Separate each custom ID with a line break.
   > - If the */etc/ilogtail/* or *C:\LogtailData* directory or the */etc/ilogtail/user_defined_id* or *C:\LogtailData\user_defined_id* file does not exist, create the directory or the file.

2. Create a machine group.
   i. Log on to the Log Service console.
   ii. In the **Projects** section, click a project.
   iii. In the left-side navigation pane, click the **Machine Groups** icon.
   iv. Click the icon next to **Machine Groups**, and select **Create Machine Group** from the shortcut menu.

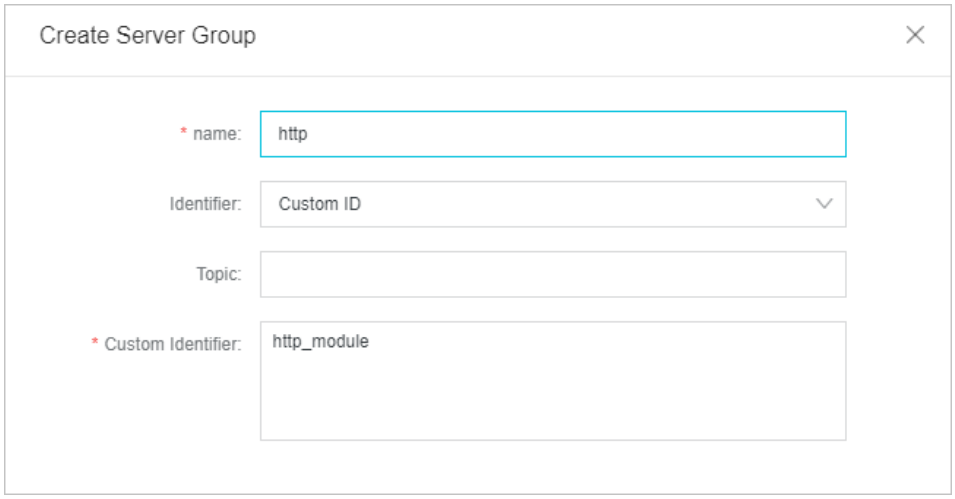
v. Set the parameters of the machine group.

- **name**: Enter a machine group name.

  The machine group name must be 3 to 128 characters in length and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit.

  > **Note** After the machine group is created, its name cannot be modified.

- **Identifier**: Select **Custom ID**.
- **Topic**: Enter a topic name for the machine group. For more information, see Generate a topic.
- **Custom Identifier**: Enter the custom ID that you set in Step 1.

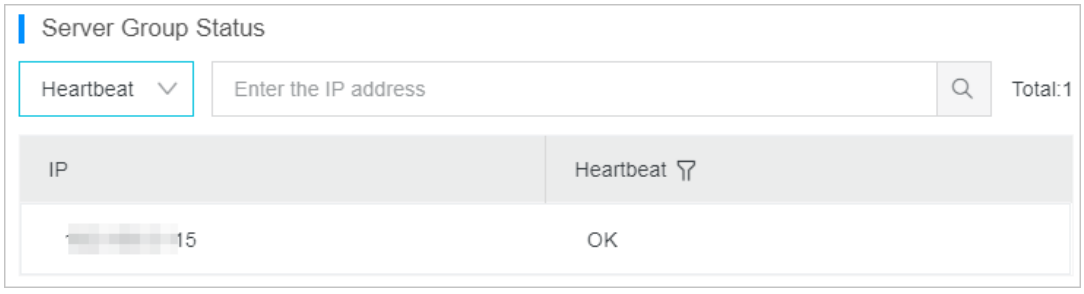


vi. Click **OK**.

> **Note** If you need to add a server to the machine group, set a custom ID of the server to the custom ID of the machine group. The server is then listed in the **Machine Group Status** section.

3. View the status of the machine group.

   In the **Machine Groups** pane, click the name of the machine group. On the **Machine Group Settings** page, view the status of the machine group. You can view the IP address list of the servers in the machine group and their heartbeat status.



> **Note**
>
> - The **Machine Group Status** section lists the IP addresses of the servers whose custom ID is the same as the custom ID that you set for the machine group.
>
> For example, the custom ID of a machine group is userdefined and the IP addresses in the **Machine Group Status** section are 10.10.10.10, 10.10.10.11, and 10.10.10.12. This means that the custom ID userdefined is set for the three servers. If you need to add the server whose IP address is 10.10.10.13 to the machine group, set the custom ID userdefined for the server. Then, the IP address of the server is displayed in the **Machine Group Status** section. Log Service collects logs from the server based on the Logtail configuration file of the machine group.
> - The heartbeat status indicates whether the connection between a server and Log Service is normal. For information about how to troubleshoot heartbeat errors, see What can I do if no heartbeat packet is received from a Logtail client?

Delete the custom IDs of a server

If you need to change the ID of a server from a custom ID to the server IP address, delete the user_defined_id file. The change takes effect within 1 minute.

- In Linux, run the following command to delete the file:

```
rm -f /etc/ilogtail/user_defined_id
```

- In Windows, run the following command to delete the file:

```
del C:\LogtailData\user_defined_id
```

Time required for a change to take effect

After you create, edit, or delete a user_defined_id file, the change takes effect within 1 minute.

If you need the change to take immediate effect, restart Logtail.

- In Linux, run the following commands to restart Logtail:

```
/etc/init.d/ilogtaild stop
/etc/init.d/ilogtaild start
```

- In Windows, perform the following steps to restart Logtail:

  Open the Control Panel. In the window that appears, choose **Control Panel > Administrative Tools > Services**. Right-click **LogtailWorker**, and select **Restart** from the shortcut menu.

Example

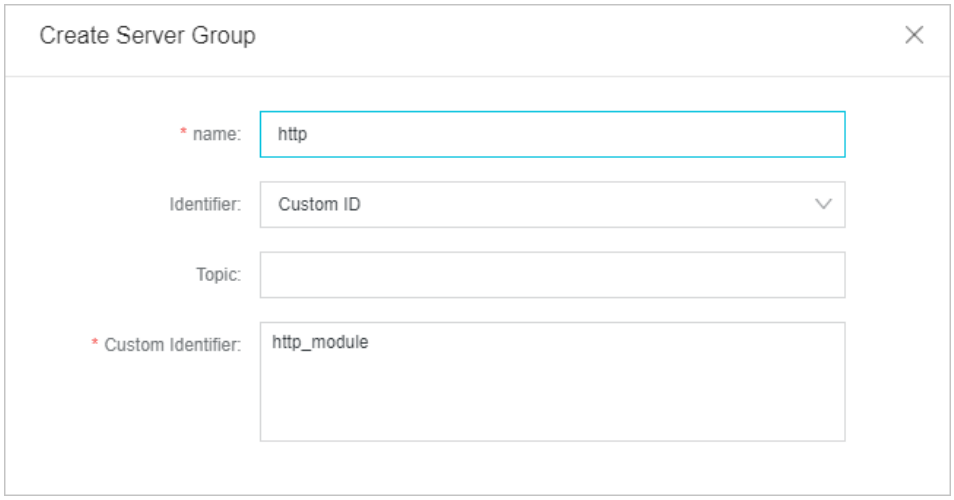
An application consists of multiple modules. Each module runs on multiple servers. For example, a website consists of an HTTP request processor, a cache, a logic processor, and a storage. You may scale out each module by adding multiple servers. You need to collect logs from both the existing and new servers.

1. Set a custom ID for each server.

   Install Logtail on the servers and set a custom ID for each server. In this example, you can use four custom IDs: http_module, cache_module, logic_module, and store_module. Each custom ID corresponds to a module.

2. Create a machine group for each module.

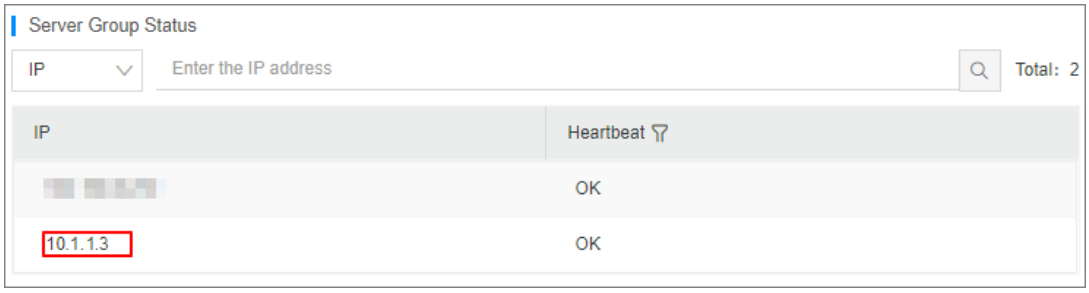
   When you create a machine group for a module, enter the custom ID of the module in the **Custom Identifier** field.



3. View the status of the machine group.

   On the **Machine Group Settings** page of the machine group, you can view the status of the machine group in the **Machine Group Status** section. You can view the list of servers in the machine group and their heartbeat status.

4. If you need to add a server whose IP address is 10.1.1.3 to the machine group whose custom ID is http_module, set the custom ID http_module for the server. Then, you can view the server in the **Machine Group Status** section.



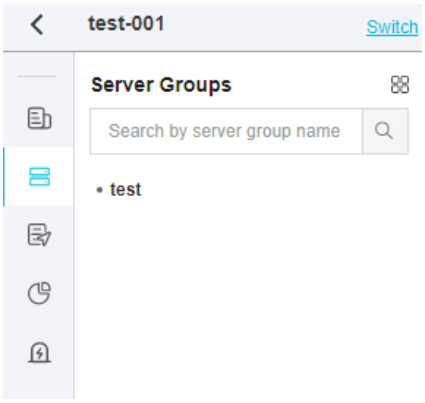
28.1.3.1.3.4. View server groups

This topic describes how to view the server groups of a project on the **Server Groups** page in the Log Service console.

Procedure

1. Log on to the Log Service console.
2. Find the target project in the project list and click the project name.
3. In the left-side navigation pane of the page that appears, click the **Server Groups** icon to display the list of server groups.

   You can view all server groups of the project.



28.1.3.1.3.5. Modify a server group

This topic describes how to modify a server group in the Log Service console. After you create a server group, you can modify the parameters of the server group.

Procedure

1. Log on to the Log Service console.
2. Find the target project in the project list and click the project name.
3. In the left-side navigation pane of the page that appears, click the **Server Groups** icon to display the list of server groups.
4. Click the name of the server group to be modified. On the **Server Group Settings** page, click **Modify**.

   > **Note** The name of the server group cannot be modified.

5. Modify the parameters of the server group, and then click **Save**.

28.1.3.1.3.6. View the status of a server group

This topic describes how to view the status of a server group in the Log Service console. You can view the heartbeat information of Logtail to check whether Logtail is installed on the servers in a server group.

Procedure

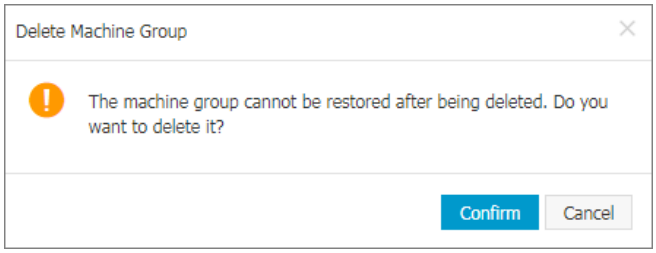
1. Log on to the Log Service console.
2. Find the target project in the project list and click the project name.
3. In the left-side navigation pane of the page that appears, click the **Server Groups** icon to display the list of server groups.
4. Click the name of the server group. On the **Server Group Settings** page, check the server group status.
   - If the heartbeat is OK, Logtail is installed on the servers in the server group and Logtail is connected to Log Service.
   - If the heartbeat status is FAIL, Logtail fails to connect to Log Service. If the FAIL state persists, perform troubleshooting based on the instructions provided in What can I do if no heartbeat packet is received from a Logtail client?

28.1.3.1.3.7. Delete a server group

This topic describes how to delete a server group in the Log Service console. You can delete a server group if you no longer need to collect logs from the server group.

Procedure

1. Log on to the Log Service console.
2. Find the target project in the project list and click the project name.
3. In the left-side navigation pane of the page that appears, click the **Server Groups** icon to display the list of server groups.
4. Find the server group that you want to delete, click the icon next to the server group, and then select **Delete**.
5. In the dialog box that appears, click **OK**.



28.1.3.1.3.8. Manage server group configurations

This topic describes how to manage server group configurations in the Log Service console. Log Service uses server groups to manage the servers from which you collect logs by using Logtail. In the Log Service console, you can create, view, modify, and delete server groups. You can also view the status of server groups, manage server group configurations, and apply server group identifiers.

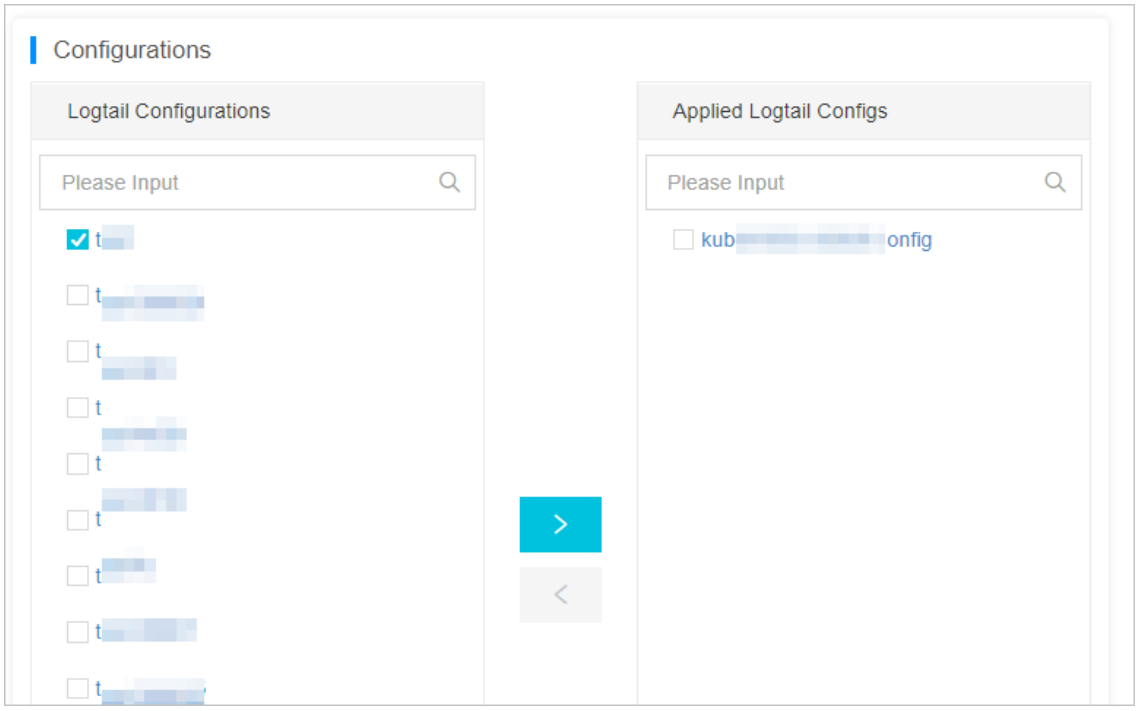
Context

Log Service allows you to manage the Logtail configurations that you create for Logtail installed on the servers in a server group. You can apply Logtail configurations to a server group. The Logtail configurations determine what logs are collected on each server, how the logs are parsed, and which Logstore the logs are written to.

Procedure

1. Log on to the Log Service console.
2. Find the target project in the project list and click the project name.
3. In the left-side navigation pane of the page that appears, click the **Server Groups** icon to display the list of server groups.
4. Click the name of the server group whose configurations you want to modify. On the **Server Group Settings** page, click **Modify**.
5. In the **Configurations** section, modify the Logtail configuration that you want to apply to the server group and click **Save**.

   After a Logtail configuration is added, it is delivered to Logtail on each server in the server group. After a Logtail configuration is removed, it is removed from Logtail.



28.1.3.1.3.9. Manage a Logtail configuration

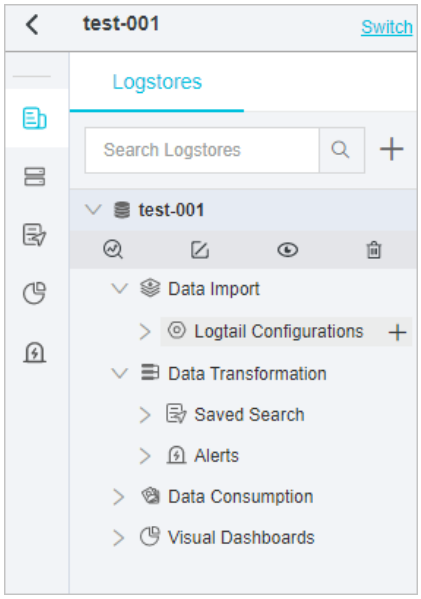
This topic describes how to manage a Logtail configuration in the Log Service console. Before you can collect logs from a server, you must install Logtail on the server. After you install Logtail, you must create a Logtail configuration in the Log Service console and apply the Logtail configuration to the server. You can create and modify Logtail configurations in Logstores.

Create a Logtail configuration

For information about how to create a Logtail configuration in the Log Service console, see Configure text log collection.

View Logtail configurations

1. Log on to the Log Service console.
2. Find the target project in the project list and click the project name.
3. In the left-side navigation pane, click the closing angle bracket (>) next to the target Logstore and choose **Data Import > Logtail Configurations**. Each item under **Logtail Configurations** indicates a Logtail configuration.



Modify a Logtail configuration

Under **Logtail Configurations**, click the name of the Logtail configuration. On the **Logtail Config** page, click **Modify**.

You can also change the log collection mode of the Logtail configuration, and then apply the Logtail configuration to the server group again. The process of modifying a Logtail configuration is the same as the process of creating a Logtail configuration.

Delete a Logtail configuration

Click the icon next to the Logtail configuration, and then select **Delete**.

After the Logtail configuration is deleted, it is disassociated from the server group. Logtail no longer collects logs specified by the Logtail configuration.

28.1.3.1.3.10. Configure an account ID on a server

This topic describes how to configure the ID of an Apsara Stack tenant account on a server.

Prerequisites

Logtail is installed on the server. For more information, see Install Logtail in Linux or Install Logtail in Windows.

Context

You must configure the ID if the server is an on-premises server or an ECS instance that belong to another Apsara Stack tenant account. You must also configure the ID if the server is a cloud server that is provided by a third-party vendor. Then, the Apsara Stack tenant account can use Logtail to collect logs from the server. If you do not configure the account ID on the server, Log Service cannot receive the heartbeat of the server and Logtail cannot collect logs from the server.

Procedure

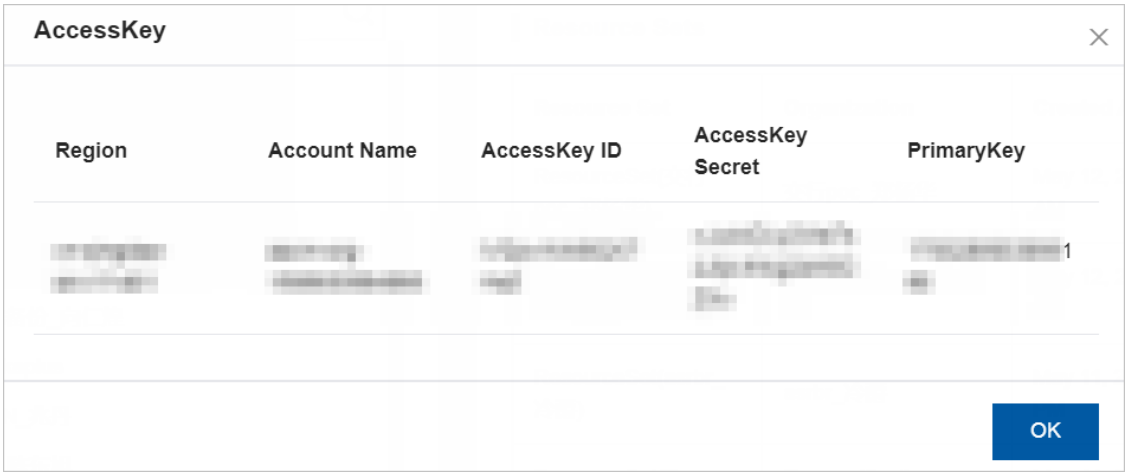
1. View the ID of your Apsara Stack tenant account.
   i. Log on to the Apsara Uni-manager Management Console.

      For more information, see Log on to the Log Service console.
   ii. In the top navigation bar, click **Enterprise**.

iii. In the left-side navigation pane, click **Organizations**.

iv. Select the destination account and click **Obtain an accesskey**

v. In the **AccessKey** dialog box, view the account ID.



2. Log on to the server and configure the account ID on the server.
   - Linux server:

     In the */etc/ilogtail/users* directory, create a file. Set the name of the file to the account ID. If the directory does not exist, create the directory first. You can configure multiple account IDs for a server. Examples:

     ```
     touch /etc/ilogtail/users/1**************
     touch /etc/ilogtail/users/1**************
     ```

     If you no longer need to collect logs from the server to a Log Service project, run the following command to delete the account ID:

     ```
     rm /etc/ilogtail/users/1**************
     ```

   - Windows server:

     In the *C:\LogtailData\users* directory, create a file. Set the name of the file to the account ID. To delete the account ID, delete the file.

     For example, you can delete the file of an account ID from the *C:\LogtailData\users\1**************** directory.

     > **Note**
     > - After you configure the ID of an Apsara Stack tenant account on a server, the account is authorized to use Logtail to collect logs from the server. If an account is no longer used to collect logs from the server, delete the account ID file from the server at the earliest opportunity.
     > - After you configure or delete an account ID, the change takes effect within 1 minute.

28.1.3.1.4. Text logs

28.1.3.1.4.1. Configure text log collection

This topic describes how to configure Logtail in the Log Service console to collect text logs from specified servers.

Prerequisites

Logtail is installed. Logtail can be installed on a Windows or Linux operating system. For more information, see Install Logtail in Linux and Install Logtail in Windows.

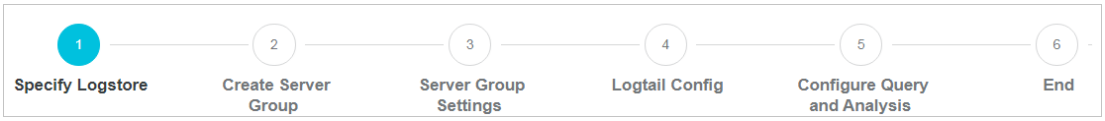
Limits

- Each log file can be collected by using only one Logtail configuration. If you want to collect a log file by using more than one Logtail configuration, we recommend that you use symbolic links. For example, to collect a log file by using two Logtail configurations in the `/home/log/nginx/log` directory, you can use the original log path for one Logtail configuration. Then, run the `ln -s /home/log/nginx/log /home/log/nginx/link_log` command to create a symbolic link for this directory and use the symbolic link as the log path for the other Logtail configuration.
- Logtail supports only Windows or Linux operating systems. For more information, see Logtail overview.

Logtail configuration procedure

You can specify Logtail configurations in the Log Service console. Logtail supports various collection modes, such as simple mode, NGINX configuration mode, Apache configuration mode, IIS configuration mode, delimiter mode, JSON mode, and full regex mode.

Configuration procedure



Collection modes

Logtail supports various collection modes, such as simple mode, NGINX configuration mode, Apache configuration mode, IIS configuration mode, delimiter mode, JSON mode, and full regex mode.

- Simple mode

  Logtail can be used to collect logs in the simple mode. For more information, see Collect logs by line.
- Full regex mode

  Logtail can be used to collect logs in the full regex mode. For more information, see Use regular expressions to collect logs.
- Delimiter mode

  Logtail can be used to collect logs in the delimiter mode. For more information, see Collect DSV formatted logs.
- JSON mode

  Logtail can be used to collect logs in the JSON mode. For more information, see Collect JSON logs.
- NGINX configuration mode

  Logtail can be used to collect logs in the NGINX configuration mode. For more information, see Collect NGINX logs.
- IIS configuration mode

  Logtail can be used to collect logs in the IIS configuration mode. For more information, see Collect IIS logs.
- Apache configuration mode

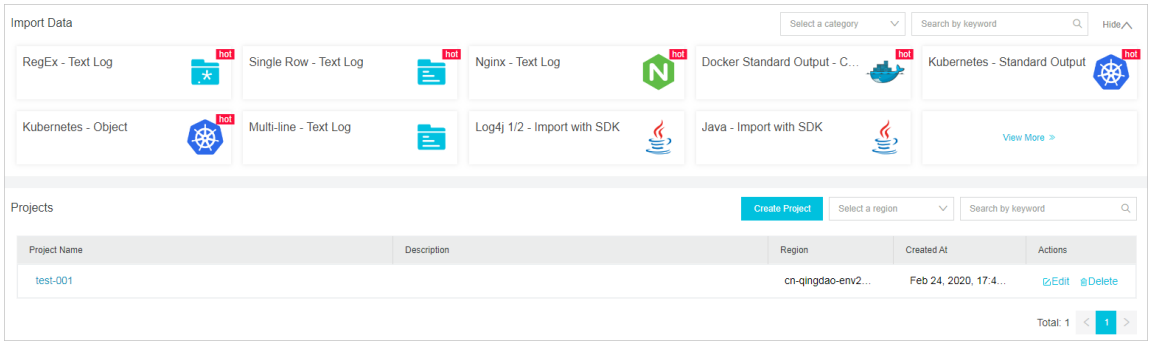
  Logtail can be used to collect logs in the Apache configuration mode. For more information, see Collect logs in Apache mode.

Procedure

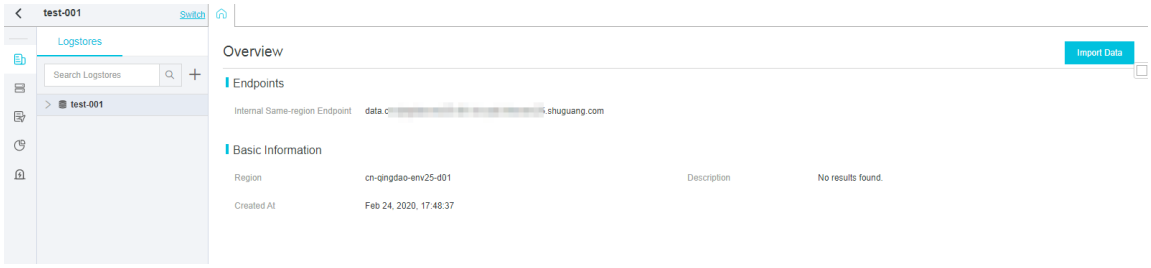
1. Log on to the Log Service console.
2. Select a data source.

   You can use one of the following three methods to select a data source:

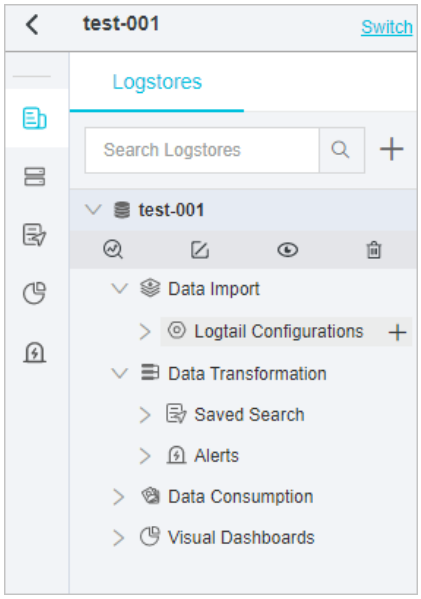
- On the homepage of the Log Service console, select a data source in the **Import Data** section.



- In the **Projects** section, click a project name. On the **Overview** page, click **Import Data**.



- On the Logstores tab in the left-side navigation pane, find a Logstore and click the closing angle bracket (>) in front of the Logstore name. Then, click the plus sign (+) next to **Data Import**.



Select a data source based on your business requirements. Log Service supports the following log sources of text logs: **RegEx-Text Log**, **Single Row-Text Log**, **Multi-Row-Text Log**, **Delimiter Mode-Text Log**, **JSON-Text Log**, **Nginx-Text Log**, **IIS-Text Log**, and **Apache-Text Log**.

3. Select a Logstore, and then click Next.

   Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see Manage a Logstore.

   If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

4. Create a server group.

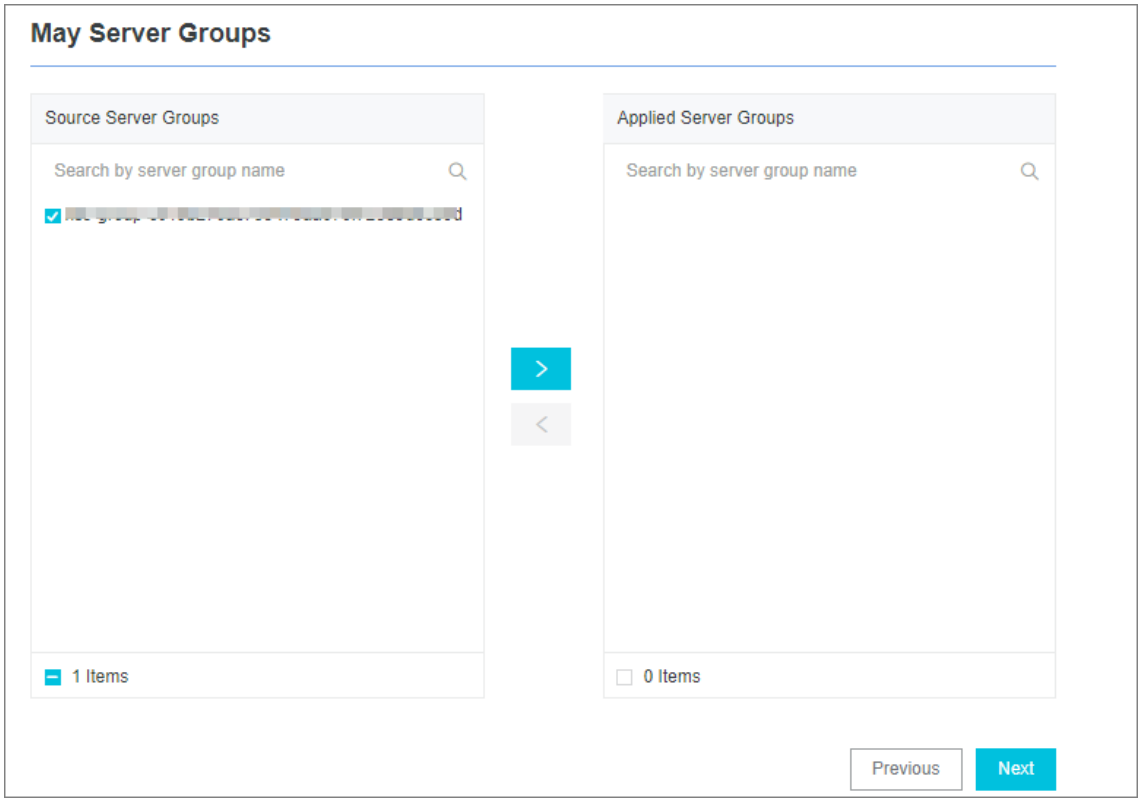
   Before you create a server group, ensure that Logtail is installed.

Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

After you install Logtail, click **Complete Installation** to create a server group. For more information, see Overview. If you have created a server group, click **Use Existing Server Groups** to select the server group.

5. Configure the server group, and then click Next.

   Select a server group and move the group from **Source Server Groups** to **Applied Server Groups**.



6. Specify Logtail parameters.

   Logtail parameters vary based on collection modes. For more information, see the relevant parameters for specific collection modes.

7. (Optional)Specify **Advanced Options** and click **Next**.

   Specify **Advanced Options** based on your business requirements. We recommend that you do not modify the default settings unless otherwise required.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs.
Upload Raw Log	Specifies whether to upload raw logs. If you turn on this switch, raw logs are written to the __raw__ field and uploaded with the parsed logs.
Topic Generation Mode	◦ **Null - Do not generate topic**: This mode is selected by default. In this mode, the topic is set to an empty string and you can query logs without the need to enter a topic. ◦ **Server Group Topic Attributes**: This mode is used to differentiate log data that is generated by different frontend servers. ◦ **File Path RegEx**: If you select this mode, you must enter a value in the **Custom RegEx** field to extract part of the path as the topic. This mode is used to differentiate log data that is generated by users or instances.

Parameter	Description
Custom RegEx	Specifies a custom regular expression. If you select **File Path RegEx** for **Topic Generation Mode**, you must enter a custom regular expression.
Log File Encoding	○ utf8: indicates UTF-8 encoding. ○ gbk: indicates GBK encoding.
Timezone	Specifies the time zone where logs are collected. ○ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ○ Custom: Select a time zone.
Timeout	If a log file is not updated within a specific period of time, Logtail considers the file to be timed out. ○ Never: All log files are continuously monitored and never time out. ○ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file.
Filter Configuration	Only logs that **meet all filter conditions** are collected. Examples: ○ Collect logs that meet a condition: Set the condition to `Key:level Regex:WARNING|ERROR`. It indicates that only logs with the severity level of WARNING or ERROR are collected. ○ Filter logs that do not meet a condition: ■ Set the condition to `Key:level Regex:^(?!.*(INFO|DEBUG)).*`. It indicates that logs with the severity level of INFO or DEBUG are not collected. ■ Set the condition to `Key:url Regex:.*^(?!.*(healthcheck)).*`. It indicates that logs whose URL contains the keyword healthcheck are not collected. For example, logs in which the key is url and the value is `/inner/healthcheck/jiankong.html` are not collected.

8. Configure an index.

   Configure an index based on your business requirements. For more information, see Enable the index feature and configure indexes for a Logstore.

After you complete the settings, you can start to collect logs.

28.1.3.1.4.2. Collect logs by line

This topic describes how to collect logs by line and configure indexes. You can specify the required settings in the Log Service console.

Context

To collect logs by line, you must select the simple mode. The simple mode can be divided into two types:

- Singleline mode

  In this mode, each line of log data is considered as a log. Two logs in a log file are separated by a line break.

Logtail does not extract log fields in this mode. The default regular expression is `(.*)`. Logtail records the system time of the current server as the timestamp of a log. You can modify or manage advanced Logtail settings after you have completed the configuration procedure. For more information, see Manage a Logtail configuration.

- Multi-line mode

  In the multi-line mode, a regular expression is used to match the first line of a log. The settings in the multi-line mode are similar to the settings in the full regex mode. For more information, see Use regular expressions to collect logs.

Procedure

1. Log on to the Log Service console.
2. Select a data source.

   Select **Single Row-Text Log**.
3. Select a destination project and Logstore, and then click **Next**.

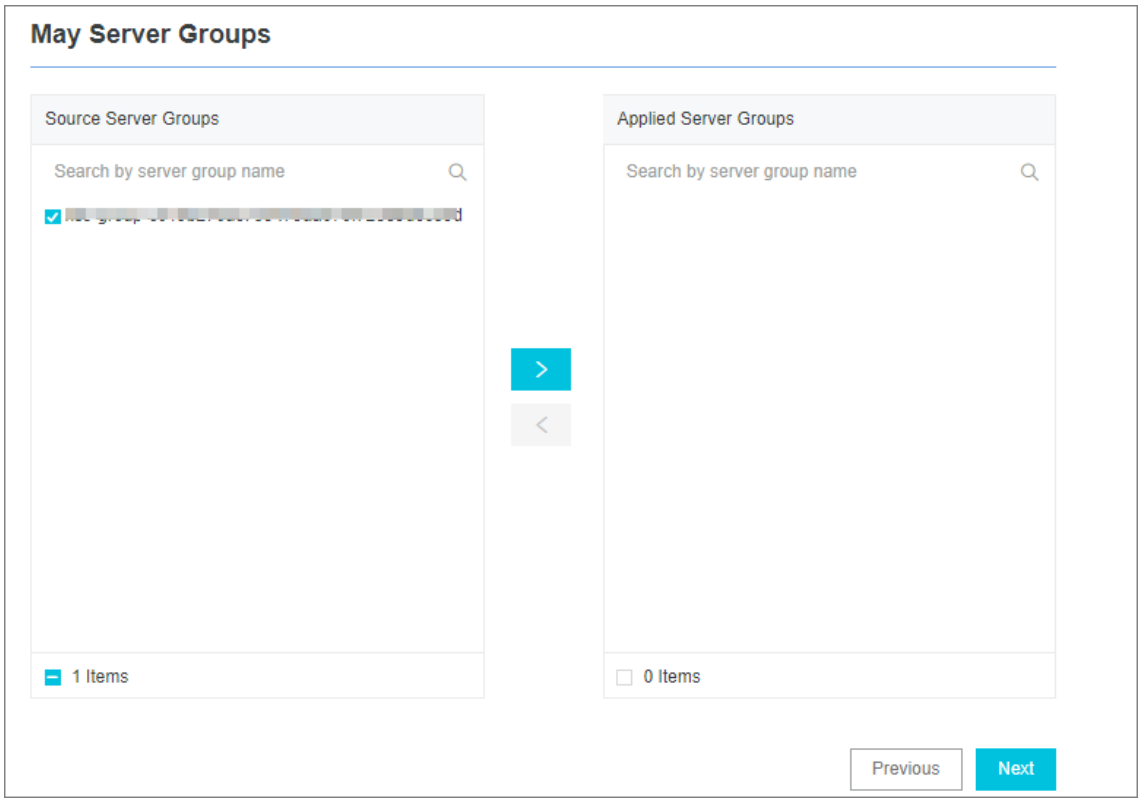
   You can also click **Create Now** to create a project and a Logstore.

   If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.
4. Create a machine group and click **Next**.

   Before you can create a machine group, you must install Logtail.

   Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

   After you install Logtail, click **Complete Installation** to create a machine group. If a machine group is created, click **Use Existing Machine Groups** to select the machine group.
5. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



> **Notice** If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

6. Create a Logtail configuration.

   The following table lists the Logtail parameters.

Parameter	Description
Config Name	The configuration name must be 3 to 128 characters in length, and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit. **Note** The configuration name cannot be modified after it is created.
Log Path	The directory and name of the log file. ○ The specified log file name can be a complete file name or a file name that contains wildcards. ○ Recursive directory matching is used in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. ■ Example 1: `/apsara/nuwa/ … /*.log` indicates the files whose extension is `.log` in the `/apsara/nuwa` directory and its sub-directories are monitored. ■ Example 2: `/var/logs/app_* … /*.log*` indicates the files whose file name contains `.log` in the following directories are monitored: the sub-directories of the `/var/logs` directory that match the `app_*` format and the sub-directories of these matching sub-directories. **Note** ■ Each log file can be collected by using only one Logtail configuration. ■ Only the asterisk (`*`) and question mark (`?`) can be used as wildcards in the log path.
Docker File	If the log file to be collected is in a Docker container, you can configure the internal path and container tag. Logtail monitors the creation and destruction of the container, filters logs of the container based on the tag, and collects the filtered logs.
Mode	If you have specified **Single Row-Text Log** for the data source, the default mode is **Simple Mode**. You can change the mode.
Maximum Directory Monitoring Depth	The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory that is specified in the log path is monitored.

7. (Optional)Specify **Advanced Options** and click **Next**.

   Set the parameters in the Advanced Options section based on your business requirements. We recommend

that you do not modify the settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs. **Note** If you turn on **Enable Plug-in Processing**, specific parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.
Upload Raw Log	If you turn on **Upload Raw Log**, each raw log is uploaded to Log Service as a value of the __raw__ field together with the parsed log.
Topic Generation Mode	The topic generation mode. ○ **Null - Do not generate topic**: This mode is selected by default. In this mode, the topic field is set to an empty string. You do not need to enter a topic to query logs. ○ **Machine Group Topic Attributes**: This mode is used to differentiate logs that are generated by different servers. ○ **File Path RegEx**: In this mode, you must specify a regular expression in the **Custom RegEx** field. The part of a log path that matches the regular expression is used as the topic. This mode is used to differentiate logs that are generated by different users or instances.
Custom RegEx	If you set the Topic Generation Mode parameter to **File Path RegEx**, you must enter a custom regular expression.
Log File Encoding	The encoding format of log files. Valid values: ○ utf8: UTF-8 encoding format ○ gbk: GBK encoding format
Timezone	The time zone where logs are collected. Valid values: ○ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ○ Custom: If you select this value, you must select a time zone.
Timeout	The timeout period of log files. If a log file is not updated within the specified period, Logtail reckons the file to be timed out. Valid values: ○ Never: All log files are continuously monitored and never time out. ○ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select **30 Minute Timeout**, you must specify the **Maximum Timeout Directory Depth** parameter. Valid values: 1 to 3.

Parameter	Description
Filter Configuration	Only logs that **meet all filter conditions** are collected. Examples: ○ Collect logs that meet specified conditions: If you set **Key** to **level** and **Regex** to **WARNING|ERROR**, only WARNING-level and ERROR-level logs are collected. ○ Filter out logs that do not meet specified conditions. ■ If you set **Key** to **level** and **Regex** to **^(?!.*(INFO|DEBUG)).***, INFO-level or DEBUG-level logs are not collected. ■ If you set **Key** to **url** and **Regex** to **.*^(?!.*(healthcheck)).***, logs whose URLs contain healthcheck are not collected. For example, the log is not collected if the value of the Key field is url and the value of the Value field is /inner/healthcheck/jiankong.html.

8. Configure indexes in the Configure Query and Analysis step. Click **Next**.

   By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Enable the index feature and configure indexes for a Logstore.

   > **Note**
   > ○ To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings of Field Search prevail.
   > ○ If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

After you complete the settings, you can start to collect logs by line.

28.1.3.1.4.3. Use regular expressions to collect logs

This topic describes how to collect logs by using regular expressions and configure indexes. You can specify the required settings in the Log Service console.

Context

If you need to collect multi-line logs and extract fields from logs, we recommend that you use regular expressions. Log Service can generate a regular expression based on a sample log that you enter in the **Import Data** wizard. However, you must modify the expression to match fields in the sample log as expected. For more information, see How do I test a regular expression?.

Procedure

1. Log on to the Log Service console.
2. Select a data source.

   Select **RegEx-Text Log**.
3. Select a destination project and Logstore, and then click **Next**.

   You can also click **Create Now** to create a project and a Logstore.

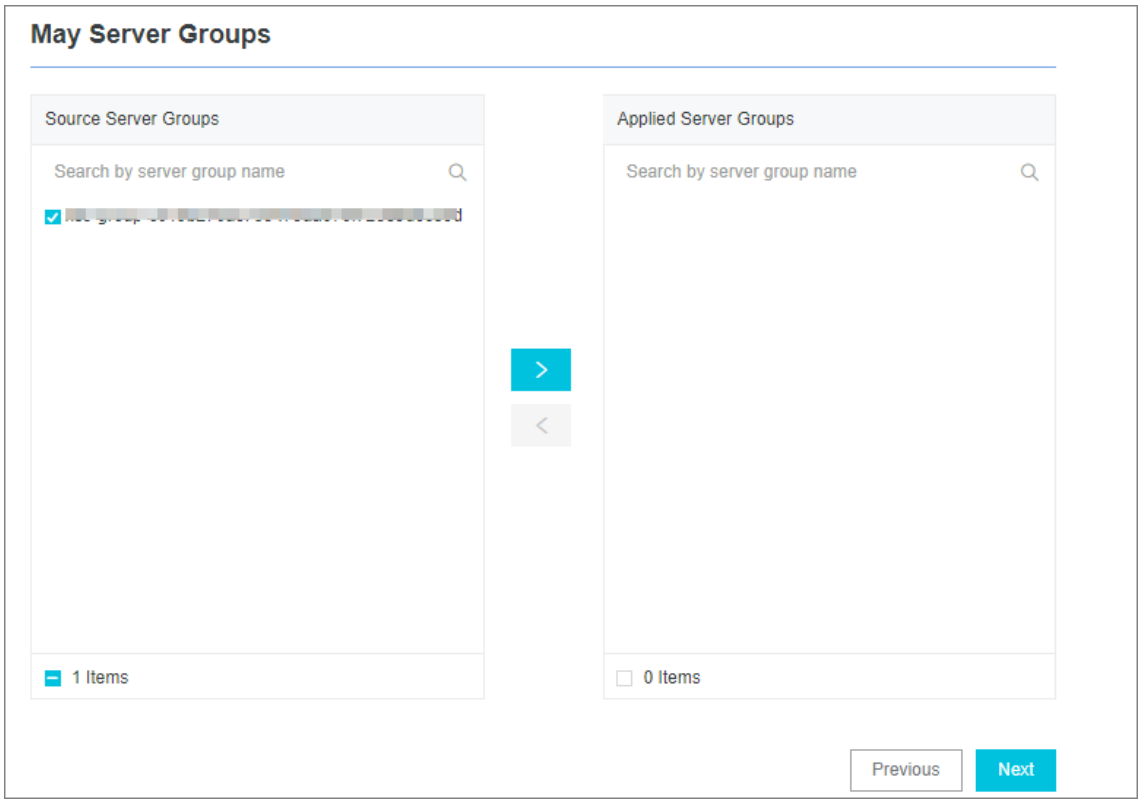
   If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.
4. Create a machine group and click **Next**.

   Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

After you install Logtail, click **Complete Installation** to create a machine group. If a machine group is created, click **Use Existing Machine Groups** to select the machine group.

5. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



> **Notice** If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

6. Create a Logtail configuration.

The following table lists the Logtail parameters.

Parameter	Description
Config Name	The configuration name must be 3 to 128 characters in length, and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit. **Note** The configuration name cannot be modified after it is created.

Parameter	Description
Log Path	The directory and name of the log file. ○ The specified log file name can be a complete file name or a file name that contains wildcards. ○ Recursive directory matching is used in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. ■ Example 1: `/apsara/nuwa/ … /*.log` indicates the files whose extension is `.log` in the `/apsara/nuwa` directory and its sub-directories are monitored. ■ Example 2: `/var/logs/app_* … /*.log*` indicates the files whose file name contains `.log` in the following directories are monitored: the sub-directories of the `/var/logs` directory that match the `app_*` format and the sub-directories of these matching sub-directories. **Note** ■ Each log file can be collected by using only one Logtail configuration. ■ Only the asterisk (`*`) and question mark (`?`) can be used as wildcards in the log path.
Docker File	If the log file to be collected is in a Docker container, you can configure the internal path and container tag. Logtail monitors the creation and destruction of the container, filters logs of the container based on the tag, and collects the filtered logs.
Mode	If you have specified **RegEx-Text Log** for the data source, the default mode is **Full Regex Mode**. You can change the mode.
Singleline	The singleline mode is enabled by default. In this mode, logs are separated by line. To collect multi-line logs, such as Java program logs, you must disable the **Singleline** mode and configure **Regex to Match First Line**.
Log Sample	Enter a sample log that is retrieved from the data source. Then, Log Service generates a regular expression.
Regex to Match First Line	You can click **Auto Generate** or **Manual**. After you enter a sample log entry and click **Auto Generate**, the system generates a regular expression. If no regular expression is generated, you can switch to the manual mode and enter a regular expression for verification.
Extract Field	To analyze and process specific fields in logs, you can turn on the **Extract Field** switch. Then, the specified fields are converted to key-value pairs and sent to Log Service. You must specify a regular expression to parse the log content.

Parameter	Description
RegEx	If you turn on the Extract Field switch, you must specify this setting. ○ Automatically generate a regular expression You can select the fields to be extracted from the sample log and then click Generate Regular Expression. The system generates a regular expression. ○ Enter a regular expression You can also enter a regular expression. Click **Manually** to switch to the manual mode. After you enter a regular expression, click **Validate** to check whether the regular expression can parse the log content. For more information, see How do I test a regular expression?.
Extracted Content	If you turn on the **Extract Field** switch, you must specify this setting. After a regular expression is automatically generated or manually specified, you must specify the key name for each extracted field.
Use System Time	If you turn on the **Extract Field** switch, you must specify this setting. If you turn off the **Use System Time** switch, you must specify a field as the time field and name this field `time`. After you specify the `time` field, click **Auto Generate** in the **Time Conversion Format** field to automatically parse the time. For more information, see Configure the time format.
Drop Failed to Parse Logs	Specifies whether to upload logs to Log Service if the logs fail to be parsed. ○ If you turn on this switch, logs that fail to be parsed are not uploaded to Log Service. ○ If you turn off this switch, raw logs are uploaded to Log Service when logs fail to be parsed.
Maximum Directory Monitoring Depth	The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory specified in the log path is monitored.

7. (Optional)Specify **Advanced Options** and click **Next**.

   Set the parameters in the Advanced Options section based on your business requirements. We recommend that you do not modify the settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs. **Note** If you turn on **Enable Plug-in Processing**, specific parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.

Parameter	Description
Upload Raw Log	If you turn on **Upload Raw Log**, each raw log is uploaded to Log Service as a value of the __raw__ field together with the parsed log.
Topic Generation Mode	The topic generation mode. ○ **Null - Do not generate topic**: This mode is selected by default. In this mode, the topic field is set to an empty string. You do not need to enter a topic to query logs. ○ **Machine Group Topic Attributes**: This mode is used to differentiate logs that are generated by different servers. ○ **File Path RegEx**: In this mode, you must specify a regular expression in the **Custom RegEx** field. The part of a log path that matches the regular expression is used as the topic. This mode is used to differentiate logs that are generated by different users or instances.
Custom RegEx	If you set the Topic Generation Mode parameter to **File Path RegEx**, you must enter a custom regular expression.
Log File Encoding	The encoding format of log files. Valid values: ○ utf8: UTF-8 encoding format ○ gbk: GBK encoding format
Timezone	The time zone where logs are collected. Valid values: ○ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ○ Custom: If you select this value, you must select a time zone.
Timeout	The timeout period of log files. If a log file is not updated within the specified period, Logtail reckons the file to be timed out. Valid values: ○ Never: All log files are continuously monitored and never time out. ○ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select **30 Minute Timeout**, you must specify the **Maximum Timeout Directory Depth** parameter. Valid values: 1 to 3.
Filter Configuration	Only logs that **meet all filter conditions** are collected. Examples: ○ Collect logs that meet specified conditions: If you set **Key** to **level** and **Regex** to **WARNING|ERROR**, only WARNING-level and ERROR-level logs are collected. ○ Filter out logs that do not meet specified conditions. ■ If you set **Key** to **level** and **Regex** to **^(?!.*(INFO|DEBUG)).***, INFO-level or DEBUG-level logs are not collected. ■ If you set **Key** to **url** and **Regex** to **.*^(?!.*(healthcheck)).***, logs whose URLs contain healthcheck are not collected. For example, the log is not collected if the value of the Key field is url and the value of the Value field is /inner/healthcheck/jiankong.html.

8. Configure indexes in the Configure Query and Analysis step. Click **Next**.

   By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Enable the index feature and configure indexes for a Logstore.

   > **Note**
   > - To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings of Field Search prevail.
   > - If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

After you complete the settings, you can start to use regular expressions to collect logs.

28.1.3.1.4.4. Collect DSV formatted logs

This topic describes how to collect delimiter-separated values (DSV) formatted logs and configure indexes. You can specify the required settings in the Log Service console.

Context

DSV formatted logs use line feeds as boundaries. Each line is a log entry. The fields of each log entry are separated with a fixed delimiter. The characters that can be used as delimiters include the tab (\t), space, vertical bar (|), comma (,), and semicolon (;). A field that contains a delimiter must be enclosed in double quotation marks (""), which are used as quotes.

Log formats

Common DSV formatted logs include comma-separated values (CSV) and tab-separated values (TSV) formatted logs.

A delimiter can contain a **single character** or **multiple characters**.

Single-character delimiter

You can specify a single-character delimiter and a quote in the console.

- **Delimiter**: The fields of each log entry are separated with a single-character delimiter, such as the tab (`\t`), vertical bar (`|`), space, comma (`,`), and semicolon (`;`). You can also specify a non-printable character as the delimiter.

  > **Note** A double quotation mark (`"`) cannot be used as a delimiter.

  If a double quotation mark (") is included in a log entry but not used as a quote, it must be escaped and processed as double quotation marks (`""`). When Log Service parse logs, it restores double quotation marks (`""`) into a double quotation mark (`"`). You can use a double quotation mark (`"`) on each boundary of a field as a quote. You can also use a double quotation mark (escaped as `""`) in the content of a field. If the use of a double quotation mark (") does not comply with the defined format, you can use the simple mode or full regex mode to parse fields.

  For example, assume that you use commas (,) as delimiters and include double quotation marks (“”) and commas (,)in a field. Enclose the field with quotes and escape the double quotation marks into `""` . For example, a processed log is `1999,Chevy,"Venture ""Extended Edition, Very Large""",,5000.00` . The log can be parsed into five fields: `1999` , `Chevy` , `Venture "Extended Edition, Very Large"` , empty field, and `5000.00` .

- **Quote**: If a log field contains delimiters, you must specify a quote to enclose the field. Otherwise, the field cannot be parsed as expected. Log Service parses the content enclosed in quotes as one field. Only delimiters

can exist between fields.

You can use one of the following characters as the quote: tab (`\t`), vertical bar (`|`), space, comma (`,`), semicolon (`;`), and non-printable characters.

For example, a log is `1997,Ford,E350,"ac, abs, moon",3000.00` . In this example, the comma (`,`) is used as the delimiter and the double quotation mark (") is used as the quote. The log entry can be parsed into five fields: `1997` , `Ford` , `E350` , `ac, abs, moon` , and `3000.00` . Among the five fields, `ac, abs, moon` enclosed in quotes is regarded as one field.

Note Log Service allows you to use a non-printable character as a delimiter or quote. Non-printable characters are characters whose decimal ASCII codes are within the range of 1 to 31 and 127. If you use a non-printable character as a delimiter or quote, you must find the hexadecimal ASCII code of this character and enter the character in the following format: `0x`*the hexadecimal ASCII code of the non-printable character* . For example, to use the non-printable character whose decimal ASCII code is 1 and hexadecimal ASCII code is 01, you must enter `0x01` .

Multi-character delimiter

Each **multi-character delimiter** can contain two or three characters, such as `||` , `&&&` , and `^_^` . If you specify a multi-character delimiter, Log Service parses logs only based on the delimiter. You do not need to use quotes to enclose log fields.

Note You must ensure that log fields do not contain the delimiter. Otherwise, Log Service cannot parse these fields as expected.

For example, if the delimiter is set to `&&` , the log `1997&&Ford&&E350&&ac&abs&moon&&3000.00` is parsed into five fields: `1997` , `Ford` , `E350` , `ac&abs&moon` , and `3000.00` .

Sample logs

- **Single-character delimiter**

```
05/May/2016:13:30:28,10.10. *. *,"POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=**********
******&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=********************
************ HTTP/1.1",200,18204,aliyun-sdk-java
05/May/2016:13:31:23,10.10. *. *,"POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=**********
******&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=********************
************ HTTP/1.1",401,23472,aliyun-sdk-java
```

- **Multi-character delimiter**

```
05/May/2016:13:30:28&&10.200. **. **&&POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=******
**********&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2Bm
kd6x7hAgQ7b1c%3D HTTP/1.1&&200&&18204&&aliyun-sdk-java
05/May/2016:13:31:23&&10.200. **. **&&POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=******
**********&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=****************
**************** HTTP/1.1&&401&&23472&&aliyun-sdk-java
```

Procedure

1. Log on to the Log Service console.
2. Select a data source.

   Select **Delimiter-Text Log**.
3. Select a destination project and Logstore, and then click **Next**.

   You can also click **Create Now** to create a project and a Logstore.

If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

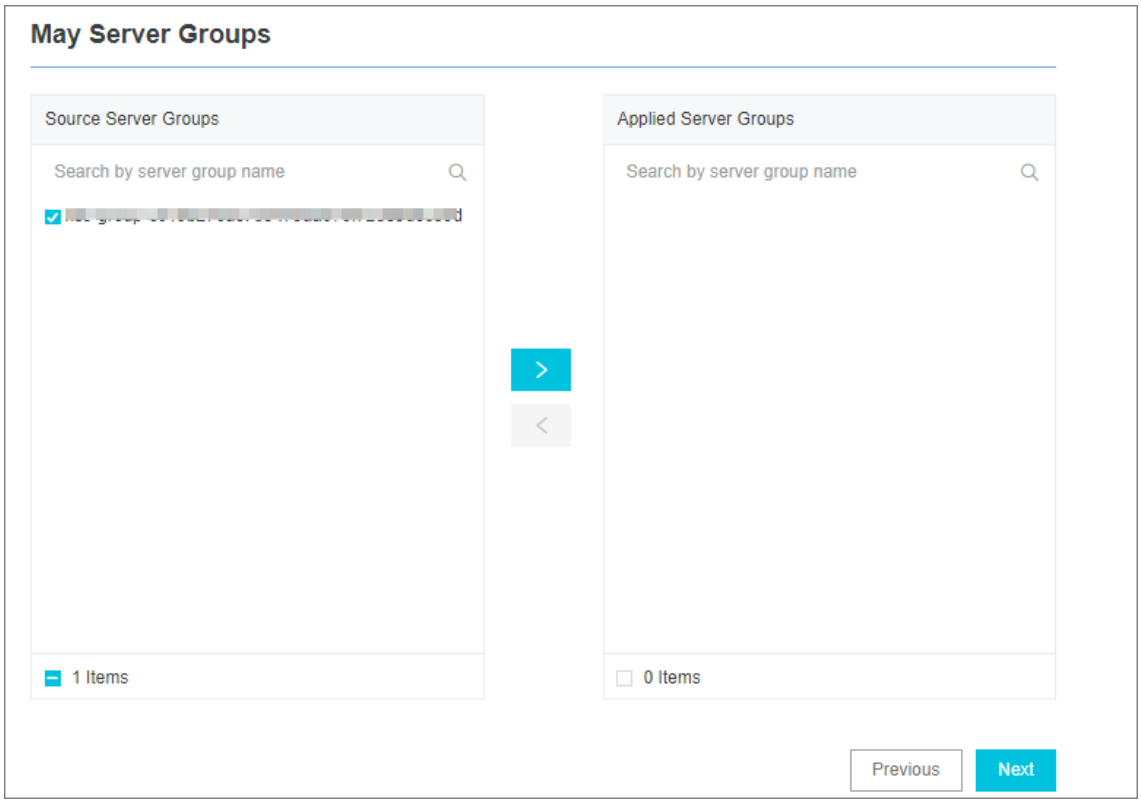
4. Create a machine group and click **Next**.

   Before you can create a machine group, you must install Logtail.

   Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

   After you install Logtail, click **Complete Installation** to create a machine group. If a machine group is created, click **Use Existing Machine Groups** to select the machine group.

5. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



> **Notice** If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

6. Create a Logtail configuration file.

   The following table lists the Logtail parameters.

Parameter	Description
Config Name	The name must be 3 to 128 characters in length, and can contain only lowercase letters, digits, hyphens (-), and underscores (_). The name must start and end with a lowercase letter or digit. **Note** The configuration name cannot be modified after it is created.

Parameter	Description
Log Path	The directory and name of the log file. ○ The specified log file name can be a complete file name or a file name that contains wildcards. ○ Recursive directory matching is used in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. ■ Example 1: `/apsara/nuwa/ … /*.log` indicates the files whose extension is `.log` in the `/apsara/nuwa` directory and its sub-directories are monitored. ■ Example 2: `/var/logs/app_* … /*.log*` indicates the files whose file name contains `.log` in the following directories are monitored: the sub-directories of the `/var/logs` directory that match the `app_*` format and the sub-directories of these matching sub-directories. **Note** ■ Each log file can be collected by using only one Logtail configuration file. ■ Only the asterisk (`*`) and question mark (`?`) can be used as wildcards in the log path.
Docker File	If you collect logs from Docker containers, you can configure the paths and tags of the containers. Logtail monitors the containers when they are created and destroyed, filters the logs of the containers by tag, and collects the filtered logs.
Mode	If you have specified **Delimiter-Text Log** for the data source, the default mode is **Delimiter Mode**. You can change the mode.
Log Sample	Enter a sample log entry that is retrieved from a log source in an actual scenario. Then, Log Service extracts a regular expression from the log entry.
Delimiter	Select a delimiter. Select a delimiter based on the log format. Otherwise, logs may fail to be parsed. **Note** If you use a non-printable character as a delimiter, you must find the hexadecimal ASCII code of this character and enter the character in the following format: `0x`*the hexadecimal ASCII code of the non-printable character*. For example, to use the non-printable character whose decimal ASCII code is 1 and hexadecimal ASCII code is 01, you must enter `0x01`.

Parameter	Description
Quote	Select a quote. Select a quote based on the log format. Otherwise, logs may fail to be parsed. **Note** If you use a non-printable character as a quote, you must find the hexadecimal ASCII code of this character and enter the character in the following format: `0x`*the hexadecimal ASCII code of the non-printable character*. For example, to use the non-printable character whose decimal ASCII code is 1 and hexadecimal ASCII code is 01, you must enter `0x01`.
Extracted Content	After you enter a sample log and select a delimiter, Log Service extracts log fields based on the delimiter and defines the fields as values. You must specify a key for each value.
Incomplete Entry Upload	This feature specifies whether to upload a log entry whose number of parsed fields is less than the number of the specified keys. If you turn on this switch, the log entry is uploaded. Otherwise, the log entry is dropped. For example, if you set the delimiter to the vertical bar (`|`), the log entry `11|22|33|44|55` can be parsed into the following fields: `11`, `22`, `33`, `44`, and `55`. You can set the keys to `A`, `B`, `C`, `D`, and `E`. If you turn on the **Incomplete Entry Upload** switch, the `55` field is uploaded as the value of the `D` key when Log Service collects the log entry `11|22|33|55`. If you turn off the **Incomplete Entry Upload** switch, Log Service drops the log entry because the fields and keys do not match.
Use System Time	If you turn on the Extract Field switch, you must specify this setting. If you turn off the Use System Time switch, you must specify a field as the time field and name this field `time`. After you specify the `time` field, click **Auto Generate** in the **Time Conversion Format** field to automatically parse the time. For more information, see Configure the time format.
Drop Failed to Parse Logs	Specifies whether to upload logs to Log Service if the logs fail to be parsed. ○ If you turn on this switch, logs that fail to be parsed are not uploaded to Log Service. ○ If you turn off this switch, raw logs are uploaded to Log Service when logs fail to be parsed.
Maximum Directory Monitoring Depth	The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory that is specified in the log path is monitored.

7. (Optional)Specify **Advanced Options** and click **Next**.

   Set the parameters in the Advanced Options section based on your business requirements. We recommend that you do not modify the settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs. **Note** If you turn on **Enable Plug-in Processing**, specific parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.
Upload Raw Log	If you turn on **Upload Raw Log**, each raw log is uploaded to Log Service as a value of the __raw__ field together with the parsed log.
Topic Generation Mode	The topic generation mode. ○ **Null - Do not generate topic**: This mode is selected by default. In this mode, the topic field is set to an empty string. You do not need to enter a topic to query logs. ○ **Machine Group Topic Attributes**: This mode is used to differentiate logs that are generated by different servers. ○ **File Path RegEx**: In this mode, you must specify a regular expression in the **Custom RegEx** field. The part of a log path that matches the regular expression is used as the topic. This mode is used to differentiate logs that are generated by different users or instances.
Custom RegEx	If you set the Topic Generation Mode parameter to **File Path RegEx**, you must enter a custom regular expression.
Log File Encoding	The encoding format of log files. Valid values: ○ utf8: UTF-8 encoding format ○ gbk: GBK encoding format
Timezone	The time zone where logs are collected. Valid values: ○ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ○ Custom: If you select this value, you must select a time zone.
Timeout	The timeout period of log files. If a log file is not updated within the specified period, Logtail reckons the file to be timed out. Valid values: ○ Never: All log files are continuously monitored and never time out. ○ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select **30 Minute Timeout**, you must specify the **Maximum Timeout Directory Depth** parameter. Valid values: 1 to 3.

Parameter	Description
Filter Configuration	Only logs that **meet all filter conditions** are collected. Examples: ○ Collect logs that meet specified conditions: If you set **Key** to **level** and **Regex** to **WARNING|ERROR**, only WARNING-level and ERROR-level logs are collected. ○ Filter out logs that do not meet specified conditions. ■ If you set **Key** to **level** and **Regex** to **^(?!.*(INFO|DEBUG)).***, INFO-level or DEBUG-level logs are not collected. ■ If you set **Key** to **url** and **Regex** to **.*^(?!.*(healthcheck)).***, logs whose URLs contain healthcheck are not collected. For example, the log is not collected if the value of the Key field is url and the value of the Value field is /inner/healthcheck/jiankong.html.

8. Configure indexes in the Configure Query and Analysis step. Click **Next**.

   By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Enable the index feature and configure indexes for a Logstore.

   > **Note**
   > ○ To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings of Field Search prevail.
   > ○ If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

After you complete the settings, you can start to collect DSV formatted logs.

28.1.3.1.4.5. Collect JSON logs

This topic describes how to use Logtail to collect JSON logs and configure indexes. You can specify the required settings in the Log Service console.

Context

Logtail can parse JSON objects from logs. It extracts the keys and values from the first layer of an object as the names and values of log fields. The valid data types of field values include object, array, and primitive data types such as string or number.

JSON logs can be written in the following two types of structures:

- Object: a collection of key-value pairs.
- Array: an ordered list of values.

Lines of JSON logs are separated with `\n`. Each line is extracted as a single log.

Logtail can parse only JSON logs of the object type. If you want to parse JSON logs of other types, such as JSON arrays, you must use regular expressions to extract the fields or specify the simple mode to collect logs by line.

Sample log

A sample JSON log is as follows:

{"url": "POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=U0Ujpek********&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2Bmkd6x7hAgQ7b1c%3D HTTP/1.1", "ip": "10.200.98.220", "user-agent": "aliyun-sdk-java", "request": {"status": "200", "latency": "18204"}, "time": "05/May/2016:13:30:28"}
{"url": "POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=U0Ujpek********&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2Bmkd6x7hAgQ7b1c%3D HTTP/1.1", "ip": "10.200.98.210", "user-agent": "aliyun-sdk-java", "request": {"status": "200", "latency": "10204"}, "time": "05/May/2016:13:30:29"}

Procedure

1. Log on to the Log Service console.
2. Select a data source.

   Select **JSON-Text Log**.
3. Select a destination project and Logstore, and then click **Next**.

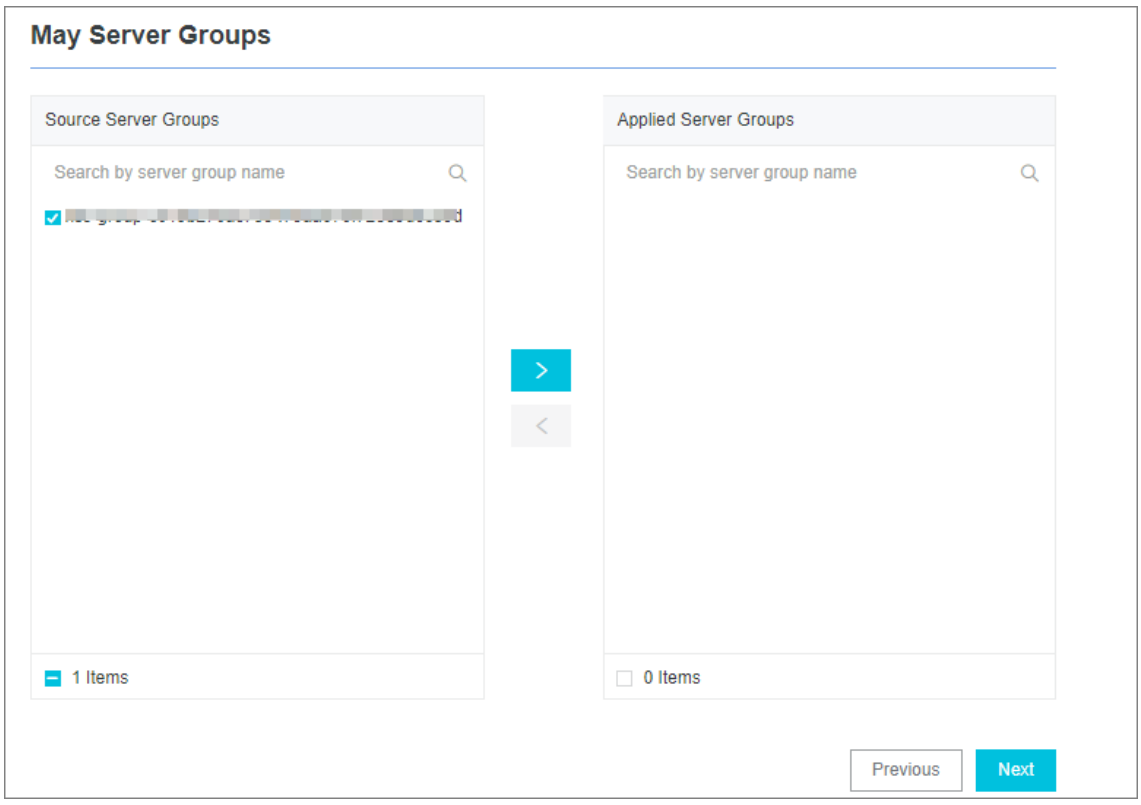
   You can also click **Create Now** to create a project and a Logstore.

   If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.
4. Create a machine group and click **Next**.

   Before you can create a machine group, you must install Logtail.

   Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

   After you install Logtail, click **Complete Installation** to create a machine group. If a machine group is created, click **Use Existing Machine Groups** to select the machine group.
5. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



> **Notice** If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

6. Create a Logtail configuration.

   The following table lists the Logtail parameters.

Parameter	Description
Config Name	The configuration name must be 3 to 128 characters in length, and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit. **Note** The configuration name cannot be modified after it is created.
Log Path	The directory and name of the log file. ○ The specified log file name can be a complete file name or a file name that contains wildcards. ○ Recursive directory matching is used in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. ■ Example 1: `/apsara/nuwa/ … /*.log` indicates the files whose extension is `.log` in the `/apsara/nuwa` directory and its sub-directories are monitored. ■ Example 2: `/var/logs/app_* … /*.log*` indicates the files whose file name contains `.log` in the following directories are monitored: the sub-directories of the `/var/logs` directory that match the `app_*` format and the sub-directories of these matching sub-directories. **Note** ■ Each log file can be collected by using only one Logtail configuration. ■ Only the asterisk (`*`) and question mark (`?`) can be used as wildcards in the log path.
Docker File	If the log file to be collected is in a Docker container, you can configure the internal path and container tag. Logtail monitors the creation and destruction of the container, filters logs of the container based on the tag, and collects the filtered logs.
Mode	If you have specified **JSON-Text Log** for the data source, the default mode is **JSON Mode**. You can change the mode.
Use System Time	If you turn on the Extract Field switch, you must specify this parameter. If you turn off the Use System Time switch, you must specify a field as the time field and name this field `time` . After you specify the `time` field, click **Auto Generate** in the **Time Conversion Format** field to automatically parse the time. For more information, see Configure the time format.

Parameter	Description
Drop Failed to Parse Logs	Specifies whether to upload logs to Log Service if the logs fail to be parsed. ○ If you turn on this switch, logs that fail to be parsed are not uploaded to Log Service. ○ If you turn off this switch, raw logs are uploaded to Log Service when logs fail to be parsed.
Maximum Directory Monitoring Depth	The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory specified in the log path is monitored.

7. (Optional)Specify **Advanced Options** and click **Next**.

   Set the parameters in the Advanced Options section based on your business requirements. We recommend that you do not modify the settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs. **Note** If you turn on **Enable Plug-in Processing**, specific parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.
Upload Raw Log	If you turn on **Upload Raw Log**, each raw log is uploaded to Log Service as a value of the __raw__ field together with the parsed log.
Topic Generation Mode	The topic generation mode. ○ **Null - Do not generate topic**: This mode is selected by default. In this mode, the topic field is set to an empty string. You do not need to enter a topic to query logs. ○ **Machine Group Topic Attributes**: This mode is used to differentiate logs that are generated by different servers. ○ **File Path RegEx**: In this mode, you must specify a regular expression in the **Custom RegEx** field. The part of a log path that matches the regular expression is used as the topic. This mode is used to differentiate logs that are generated by different users or instances.
Custom RegEx	If you set the Topic Generation Mode parameter to **File Path RegEx**, you must enter a custom regular expression.
Log File Encoding	The encoding format of log files. Valid values: ○ utf8: UTF-8 encoding format ○ gbk: GBK encoding format

Parameter	Description
Timezone	The time zone where logs are collected. Valid values: ○ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ○ Custom: If you select this value, you must select a time zone.
Timeout	The timeout period of log files. If a log file is not updated within the specified period, Logtail reckons the file to be timed out. Valid values: ○ Never: All log files are continuously monitored and never time out. ○ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select **30 Minute Timeout**, you must specify the **Maximum Timeout Directory Depth** parameter. Valid values: 1 to 3.
Filter Configuration	Only logs that **meet all filter conditions** are collected. Examples: ○ Collect logs that meet specified conditions: If you set **Key** to **level** and **Regex** to **WARNING|ERROR**, only WARNING-level and ERROR-level logs are collected. ○ Filter out logs that do not meet specified conditions. ■ If you set **Key** to **level** and **Regex** to **^(?!.*(INFO|DEBUG)).***, INFO-level or DEBUG-level logs are not collected. ■ If you set **Key** to **url** and **Regex** to **.*^(?!.*(healthcheck)).***, logs whose URLs contain healthcheck are not collected. For example, the log is not collected if the value of the Key field is url and the value of the Value field is /inner/healthcheck/jiankong.html.

8. Configure indexes in the Configure Query and Analysis step. Click **Next**.

   By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Enable the index feature and configure indexes for a Logstore.

   > **Note**
   > ○ To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings of Field Search prevail.
   > ○ If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

After you complete the settings, you can start to collect JSON logs.

28.1.3.1.4.6. Collect NGINX logs

This topic describes how to collect NGINX logs and configure indexes. You can connect Log Service to NGINX and specify the required settings in the Log Service console.

Context

The NGINX log format and path are specified in the */etc/nginx/nginx.conf* configuration file.

NGINX log format

In the configuration file, the format of NGINX logs is defined as follows:

```
log_format main  '$remote_addr - $remote_user [$time_local] "$request" '
                 '$request_time $request_length '
                 '$status $body_bytes_sent "$http_referer" '
                 '"$http_user_agent"';
```

The path of the log file is declared as follows. The "main" portion that follows the path indicates that logs are written in the preceding format.

```
access_log /var/logs/nginx/access.log main
```

Sample log

A sample NGINX log is as follows:

```
192.168.1.2 - - [10/Jul/2015:15:51:09 +0800] "GET /ubuntu.iso HTTP/1.0" 0.000 129 404 168 "-" "Wget/1
.11.4 Red Hat modified"
```

NGINX log fields

Field	Description
remote_addr	The IP address of the client.
remote_user	The username of the client.
request	The URL and HTTP protocol of the request.
status	The status of the request.
body_bytes_sent	The number of bytes in the response that is returned to the client, excluding the size of the response header.
connection	The serial number of a connection.
connection_requests	The number of requests that are received from a connection.
msec	The time when the log is written. The time is measured in seconds, accurate to milliseconds.
pipe	Indicates whether the request is pipelined. If the request is pipelined, the field value is `p`. Otherwise, the field value is `.`.
http_referer	The URL of the web page linked to the resource that is being requested.
“http_user_agent”	The browser information of the client. The information must be enclosed by double quotation marks ("").
request_length	The length of the request. The length includes the request line, request header, and request body.

Field	Description
request_time	The time period for which the request is processed. The time period is measured in seconds, accurate to milliseconds. The time period starts when the first byte is read from the client and ends when the log is written after the last byte is sent to the client.
[$time_local]	The local time in the Common Log Format. The time must be enclosed by brackets [].

Procedure

1. Log on to the Log Service console.
2. Select a data source.

   Select **Nginx-Text Log**.
3. Select a destination project and Logstore, and then click **Next**.

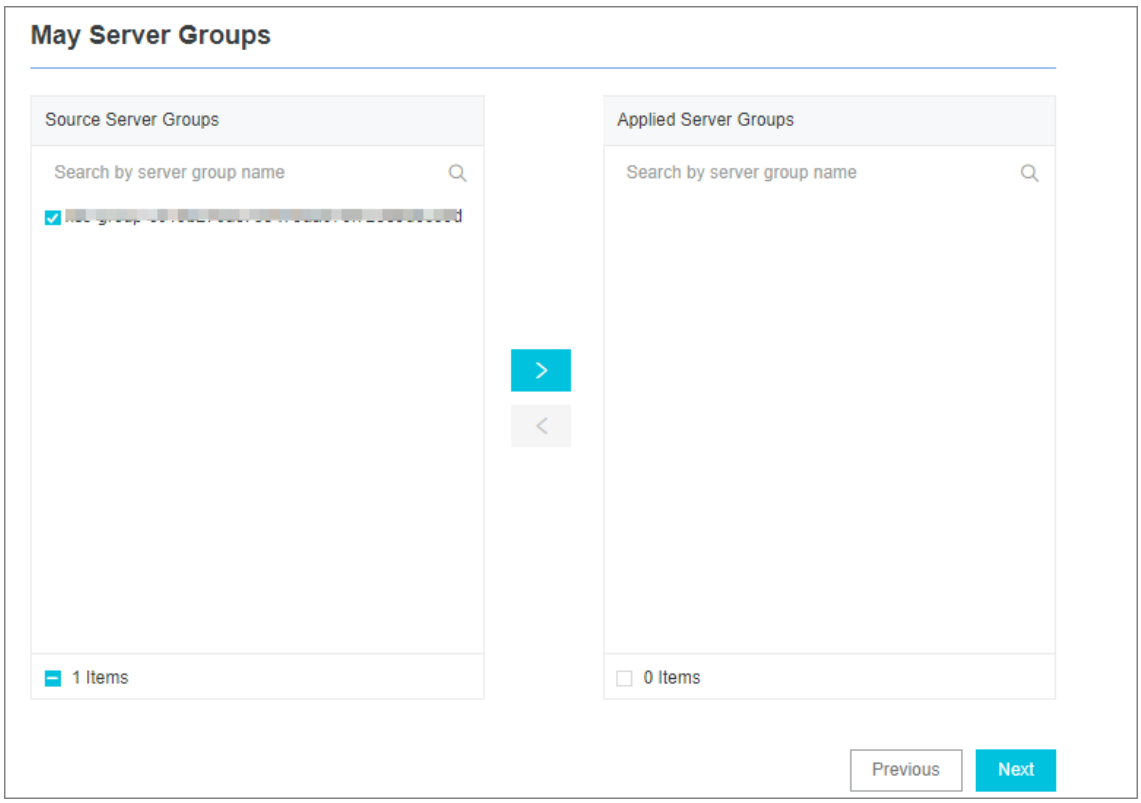
   You can also click **Create Now** to create a project and a Logstore.

   If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.
4. Create a machine group and click **Next**.

   Before you can create a machine group, you must install Logtail.

   Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

   After you install Logtail, click **Complete Installation** to create a machine group. If a machine group is created, click **Use Existing Machine Groups** to select the machine group.
5. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



> **Notice** If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

6. Create a Logtail configuration.

   The following table lists the Logtail parameters.

Parameter	Description
Config Name	The configuration name must be 3 to 128 characters in length, and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit. **Note** The configuration name cannot be modified after it is created.
Log Path	The directory and name of the log file. ○ The specified log file name can be a complete file name or a file name that contains wildcards. ○ Recursive directory matching is adopted in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. ■ Example 1: `/apsara/nuwa/ … /*.log` indicates the files whose extension is `.log` in the `/apsara/nuwa` directory and its sub-directories are monitored. ■ Example 2: `/var/logs/app_* … /*.log*` indicates the files whose file name contains `.log` in the following directories are monitored: the sub-directories of the `/var/logs` directory that match the `app_*` format and the sub-directories of these matching sub-directories. **Note** ■ Each log file can be collected by using only one Logtail configuration. ■ Only the asterisk (`*`) and question mark (`?`) can be used as wildcards in the log path.
Docker File	If the log file to be collected is in a Docker container, you can configure the internal path and container tag. Logtail monitors the creation and destruction of the container, filters logs of the container based on the tag, and collects the filtered logs.
Mode	If you have specified **Nginx-Text Log** for the data source, the default mode is **NGINX Configuration Mode**. You can change the mode.
NGINX Log Configuration	Enter the log configuration section that is specified in a standard NGINX configuration file. The section starts with `log_format`.
NGINX Key	Log Service reads the keys of NGINX logs.

Parameter	Description
Drop Failed to Parse Logs	Specifies whether to upload logs to Log Service if the logs fail to be parsed. ○ If you turn on this switch, logs that fail to be parsed are not uploaded to Log Service. ○ If you turn off this switch, raw logs are uploaded to Log Service when logs fail to be parsed.
Maximum Directory Monitoring Depth	The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory specified in the log path is monitored.

7. (Optional)Specify **Advanced Options** and click **Next**.

   Set the parameters in the Advanced Options section based on your business requirements. We recommend that you do not modify the settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs. **Note** If you turn on **Enable Plug-in Processing**, specific parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.
Upload Raw Log	If you turn on **Upload Raw Log**, each raw log is uploaded to Log Service as a value of the __raw__ field together with the parsed log.
Topic Generation Mode	The topic generation mode. ○ **Null - Do not generate topic**: This mode is selected by default. In this mode, the topic field is set to an empty string. You do not need to enter a topic to query logs. ○ **Machine Group Topic Attributes**: This mode is used to differentiate logs that are generated by different servers. ○ **File Path RegEx**: In this mode, you must specify a regular expression in the **Custom RegEx** field. The part of a log path that matches the regular expression is used as the topic. This mode is used to differentiate logs that are generated by different users or instances.
Custom RegEx	If you set the Topic Generation Mode parameter to **File Path RegEx**, you must enter a custom regular expression.
Log File Encoding	The encoding format of log files. Valid values: ○ utf8: UTF-8 encoding format ○ gbk: GBK encoding format

Parameter	Description
Timezone	The time zone where logs are collected. Valid values: ○ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ○ Custom: If you select this value, you must select a time zone.
Timeout	The timeout period of log files. If a log file is not updated within the specified period, Logtail reckons the file to be timed out. Valid values: ○ Never: All log files are continuously monitored and never time out. ○ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select **30 Minute Timeout**, you must specify the **Maximum Timeout Directory Depth** parameter. Valid values: 1 to 3.
Filter Configuration	Only logs that **meet all filter conditions** are collected. Examples: ○ Collect logs that meet specified conditions: If you set **Key** to **level** and **Regex** to **WARNING|ERROR**, only WARNING-level and ERROR-level logs are collected. ○ Filter out logs that do not meet specified conditions. ■ If you set **Key** to **level** and **Regex** to **^(?!.*(INFO|DEBUG)).***, INFO-level or DEBUG-level logs are not collected. ■ If you set **Key** to **url** and **Regex** to **.*^(?!.*(healthcheck)).***, logs whose URLs contain healthcheck are not collected. For example, the log is not collected if the value of the Key field is url and the value of the Value field is /inner/healthcheck/jiankong.html.

8. Configure indexes in the Configure Query and Analysis step. Click **Next**.

   By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Enable the index feature and configure indexes for a Logstore.

   > **Note**
   > ○ To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings of Field Search prevail.
   > ○ If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

After you complete the settings, you can start to collect NGINX logs.

28.1.3.1.4.7. Collect IIS logs

This topic describes how to collect Internet Information Services (IIS) logs and configure indexes. You can specify the required settings in the Log Service console.

Context

To meet log analysis requirements, we recommend that you use the W3C Extended Log File Format. To use this format, click **Select Fields** in the IIS Manager, and then select sc-bytes and cs-bytes in the Standard Fields list.

Log format

The W3C Extended Log File Format is as follows:

```
logExtFileFlags="Date, Time, ClientIP, UserName, SiteName, ComputerName, ServerIP, Method, UriStem, UriQuery, HttpStatus, Win32Status, BytesSent, BytesRecv, TimeTaken, ServerPort, UserAgent, Cookie, Referer, ProtocolVersion, Host, HttpSubStatus"
```

- Field prefixes

Prefix	Description
s-	The server action.
c-	The client action.
cs-	The client-to-server action.
sc-	The server-to-client action.

- Fields

Field	Description
date	The date on which the client sends the request.
time	The time when the client sends the request.
s-sitename	The Internet service name and instance number of the site visited by the client.
s-computername	The name of the server on which the log is generated.
s-ip	The IP address of the server on which the log is generated.
cs-method	The HTTP request method that is used by the client, for example, GET or POST.
cs-uri-stem	The URI resource requested by the client.
cs-uri-query	The query string that follows the question mark (?) in the HTTP request.
s-port	The port number of the server to which the client is connected.
cs-username	The username used by the client to access the server. Authenticated users are referenced as `domain\username`. Anonymous users are indicated by a hyphen (-).
c-ip	The IP address of the client that sends the request.
cs-version	The protocol version that is used by the client, for example, HTTP 1.0 or HTTP 1.1.
user-agent	The browser that is used by the client.
Cookie	The content of the sent or received cookie. A hyphen (-) is used if no cookie is sent or received.

Field	Description
referer	The site that the client last visited. This site provides a link to the current site.
cs-host	The header name of the host.
sc-status	The HTTP or FTP status code that is returned by the server.
sc-substatus	The HTTP substatus code that is returned by the server.
sc-win32-status	The Windows status code that is returned by the server.
sc-bytes	The number of bytes that are sent by the server.
cs-bytes	The number of bytes that are received by the server.
time-taken	The processing time of the request. Unit: milliseconds.

Procedure

1. Log on to the Log Service console.
2. Select a data source.

   Select **IIS-Text Log**.
3. Select a destination project and Logstore, and then click **Next**.

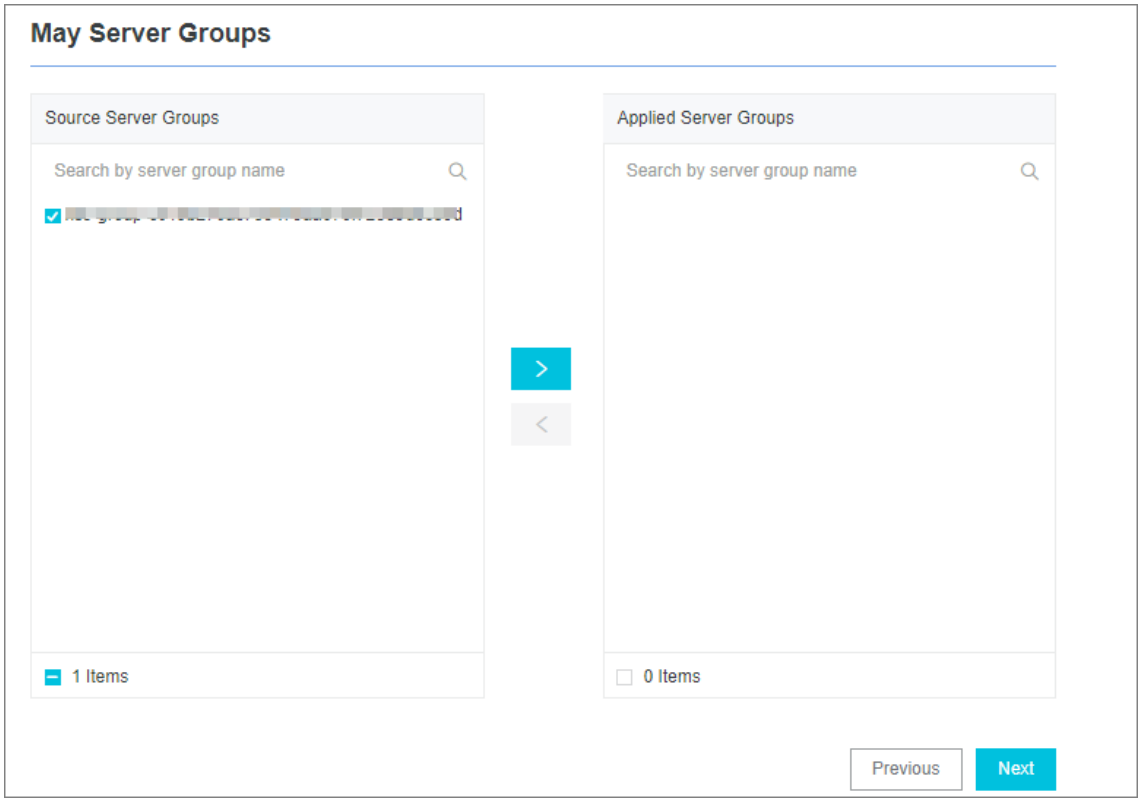
   You can also click **Create Now** to create a project and a Logstore.

   If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.
4. Create a machine group and click **Next**.

   Before you can create a machine group, you must install Logtail.

   Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

   After you install Logtail, click **Complete Installation** to create a machine group. If a machine group is created, click **Use Existing Machine Groups** to select the machine group.
5. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



Notice If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

6. Create a Logtail configuration.

   The following table lists the Logtail parameters.

Parameter	Description
Config Name	The configuration name must be 3 to 128 characters in length, and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit. **Note** The configuration name cannot be modified after it is created.

Parameter	Description
Log Path	The directory and name of the log file. ○ The specified log file name can be a complete file name or a file name that contains wildcards. ○ Recursive directory matching is used in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. ■ Example 1: `/apsara/nuwa/ … /*.log` indicates the files whose extension is `.log` in the `/apsara/nuwa` directory and its sub-directories are monitored. ■ Example 2: `/var/logs/app_* … /*.log*` indicates the files whose file name contains `.log` in the following directories are monitored: the sub-directories of the `/var/logs` directory that match the `app_*` format and the sub-directories of these matching sub-directories. **Note** ■ Each log file can be collected by using only one Logtail configuration. ■ Only the asterisk (`*`) and question mark (`?`) can be used as wildcards in the log path.
Docker File	If the log file to be collected is in a Docker container, you can configure the internal path and container tag. Logtail monitors the creation and destruction of the container, filters logs of the container based on the tag, and collects the filtered logs. For more information, see Collect container text logs.
Mode	If you have specified **IIS-Text Log** for the data source, the default mode is **IIS Configuration Mode**. You can change the mode.
Log Format	Select the log format of your IIS server logs. Valid values: ○ IIS: Microsoft IIS log file format ○ NCSA: NCSA Common log file format ○ W3C: W3C Extended Log File Format
IIS Configuration	Enter the log configuration section that is specified in an IIS configuration file. ○ If you select IIS or NCSA, the fields of the IIS log format are preconfigured. ○ If you select W3C, enter the content that is specified for the logFile logExtFileFlags in the configuration file. For more information, see Specify the IIS Configuration field.
IIS Key Name	Log Service reads the keys of IIS logs.

Parameter	Description
Drop Failed to Parse Logs	Specifies whether to upload logs to Log Service if the logs fail to be parsed. ○ If you turn on this switch, logs that fail to be parsed are not uploaded to Log Service. ○ If you turn off this switch, raw logs are uploaded to Log Service when logs fail to be parsed.
Maximum Directory Monitoring Depth	The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory that is specified in the log path is monitored.

7. Specify the IIS Configuration field.
   i. Open the IIS configuration file.
      - Default path of the IIS5 configuration file: *C:\WINNT\system32\inetsrv\MetaBase.bin*
      - Default path of the IIS6 configuration file: *C:\WINDOWS\system32\inetsrv\MetaBase.xml*
      - Default path of the IIS7 configuration file: *C:\Windows\System32\inetsrv\config\applicationHost.config*
   ii. Find the `logFile logExtFileFlags` field and copy the text in the quotation marks that follow the field name.
   iii. Paste the text into the quotation marks ("") in the **IIS Configuration** field.
8. (Optional)Specify **Advanced Options** and click **Next**.

   Set the parameters in the Advanced Options section based on your business requirements. We recommend that you do not modify the settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs. **Note** If you turn on **Enable Plug-in Processing**, specific parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.
Upload Raw Log	If you turn on **Upload Raw Log**, each raw log is uploaded to Log Service as a value of the __raw__ field together with the parsed log.
Topic Generation Mode	The topic generation mode. ○ **Null - Do not generate topic**: This mode is selected by default. In this mode, the topic field is set to an empty string. You do not need to enter a topic to query logs. ○ **Machine Group Topic Attributes**: This mode is used to differentiate logs that are generated by different servers. ○ **File Path RegEx**: In this mode, you must specify a regular expression in the **Custom RegEx** field. The part of a log path that matches the regular expression is used as the topic. This mode is used to differentiate logs that are generated by different users or instances.

Parameter	Description
Custom RegEx	If you set the Topic Generation Mode parameter to **File Path RegEx**, you must enter a custom regular expression.
Log File Encoding	The encoding format of log files. Valid values: ○ utf8: UTF-8 encoding format ○ gbk: GBK encoding format
Timezone	The time zone where logs are collected. Valid values: ○ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ○ Custom: If you select this value, you must select a time zone.
Timeout	The timeout period of log files. If a log file is not updated within the specified period, Logtail reckons the file to be timed out. Valid values: ○ Never: All log files are continuously monitored and never time out. ○ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select **30 Minute Timeout**, you must specify the **Maximum Timeout Directory Depth** parameter. Valid values: 1 to 3.
Filter Configuration	Only logs that **meet all filter conditions** are collected. Examples: ○ Collect logs that meet specified conditions: If you set **Key** to **level** and **Regex** to **WARNING|ERROR**, only WARNING-level and ERROR-level logs are collected. ○ Filter out logs that do not meet specified conditions. ■ If you set **Key** to **level** and **Regex** to **^(?!.*(INFO|DEBUG)).***, INFO-level or DEBUG-level logs are not collected. ■ If you set **Key** to **url** and **Regex** to **.*^(?!.*(healthcheck)).***, logs whose URLs contain healthcheck are not collected. For example, the log is not collected if the value of the Key field is url and the value of the Value field is /inner/healthcheck/jiankong.html.

9. Configure indexes in the Configure Query and Analysis step. Click **Next**.

   By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Enable the index feature and configure indexes for a Logstore.

   > **Note**
   > ○ To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings of Field Search prevail.
   > ○ If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

After you complete the settings, you can start to collect IIS logs.

28.1.3.1.4.8. Collect Apache logs

This topic describes how to collect Apache logs and configure indexes. You can specify the required settings in the Log Service console.

Log formats

The Apache configuration file defines two log formats: combined log format and common log format. You can also customize a log format.

- Syntax of the combined log format:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

- Syntax of the common log format:

```
LogFormat "%h %l %u %t \"%r\" %>s %b"
```

- Syntax of a custom log format:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %D %f %k %p %q %R %T %I %O
" customized
```

You must specify the log format, log file directory, and log file name in the Apache configuration file. For example, the following declaration in the configuration file indicates that the combined log format is used. The log file directory is */var/log/apache2/access_log* and the log file name is access_log.

```
CustomLog "/var/log/apache2/access_log" combined
```

Apache log fields

Format string	Key name	Description
%a	client_addr	The IP address of the client in the request.
%A	local_addr	The local private IP address.
%b	response_size_bytes	The size of the response. Unit: bytes. If no bytes are sent, the value is “-”.
%B	response_bytes	The size of the response. Unit: bytes. If no bytes are sent, the value is 0.
%D	request_time_msec	The time period for which the request is processed. Unit: milliseconds.
%h	remote_addr	The name of the remote host.
%H	request_protocol_supple	The request protocol.
%l	remote_ident	The identity information that is provided by a remote computer.
%m	request_method_supple	The request method.
%p	remote_port	The port number of the server.
%P	child_process	The ID of the child process.
%q	request_query	The query string. If it does not exist, the value is an empty string.

Format string	Key name	Description
“%r”	request	The request, which includes the method name, address, and HTTP protocol.
%s	status	The HTTP status code for the response.
%>s	status	The HTTP status code for the final response.
%f	filename	The name of the requested file.
%k	keep_alive	The number of keep-alive requests.
%R	response_handler	The type of the handler that generates the response on the server.
%t	time_local	The local time when the server receives the request.
%T	request_time_sec	The time period for which the request is processed. Unit: seconds.
%u	remote_user	The username that you used to log on to the client.
%U	request_uri_supple	The requested URL, excluding query strings.
%v	server_name	The name of the server.
%V	server_name_canonical	The server name based on the UseCanonicalName setting.
%I	bytes_received	The number of bytes that are received by the server. To use this field, you must enable the mod_logio module.
%O	bytes_sent	The number of bytes that are sent by the server. To use this field, you must enable the mod_logio module.
“%{User-Agent}i”	http_user_agent	The information about the client.
“%{Rererer}i”	http_referer	The URL of the web page linked to the resource that is being requested.

Sample log

```
192.168.1.2 - - [02/Feb/2016:17:44:13 +0800] "GET /favicon.ico HTTP/1.1" 404 209 "http://localhost/x1.html" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.97 Safari/537.36"
```

Procedure

1. Log on to the Log Service console.
2. Select a data source.

   Select **Apache-Text Log**.
3. Select a destination project and Logstore, and then click **Next**.

You can also click **Create Now** to create a project and a Logstore.

If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

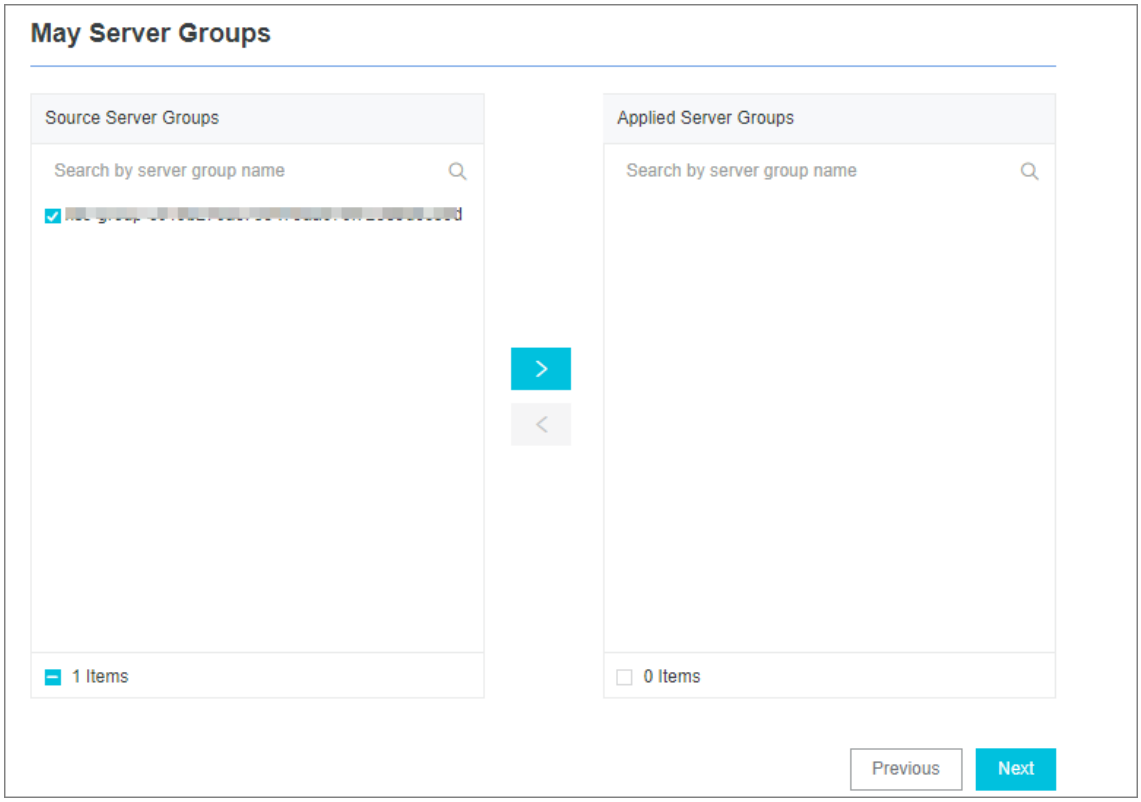
4. Create a machine group and click **Next**.

   Before you can create a machine group, you must install Logtail.

   Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

   After you install Logtail, click **Complete Installation** to create a machine group. If a machine group is created, click **Use Existing Machine Groups** to select the machine group.

5. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



Notice If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

6. Create a Logtail configuration.

   The following table lists the Logtail parameters.

Parameter	Description
Config Name	The configuration name must be 3 to 128 characters in length, and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit. **Note** The configuration name cannot be modified after it is created.

Parameter	Description
Log Path	The directory and name of the log file. ○ The specified log file name can be a complete file name or a file name that contains wildcards. ○ Recursive directory matching is used in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. ■ Example 1: `/apsara/nuwa/ … /*.log` indicates the files whose extension is `.log` in the `/apsara/nuwa` directory and its sub-directories are monitored. ■ Example 2: `/var/logs/app_* … /*.log*` indicates the files whose file name contains `.log` in the following directories are monitored: the sub-directories of the `/var/logs` directory that match the `app_*` format and the sub-directories of these matching sub-directories. **Note** ■ Each log file can be collected by using only one Logtail configuration. ■ Only the asterisk (`*`) and question mark (`?`) can be used as wildcards in the log path.
Docker File	If the log file to be collected is in a Docker container, you can configure the internal path and container tag. Logtail monitors the creation and destruction of the container, filters logs of the container based on the tag, and collects the filtered logs. For more information, see Collect container text logs.
Mode	If you have specified **Apache-Text Log** for the data source, the default mode is **Apache Configuration Mode**. You can change the mode.
Log Format	Select a log format based on the format declared in your Apache log configuration file. To facilitate the query and analysis of log data, we recommend that you use a custom Apache log format.
APACHE Logformat Configuration	Enter the log configuration section that is specified in the Apache configuration file. The section starts with LogFormat. **Note** If the specified **Log Format** is **Common** or **Combined**, the system enters a commonly used syntax of the log format. Check whether the log format is the same as that defined in the Apache configuration file.
APACHE Key Name	Log Service reads the keys of Apache logs. Confirm the key names on the Logtail configuration page.

Parameter	Description
Drop Failed to Parse Logs	Specifies whether to upload logs to Log Service if the logs fail to be parsed. ○ If you turn on this switch, logs that fail to be parsed are not uploaded to Log Service. ○ If you turn off this switch, raw logs are uploaded to Log Service when logs fail to be parsed.
Maximum Directory Monitoring Depth	The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory specified in the log path is monitored.

7. (Optional)Specify **Advanced Options** and click **Next**.

   Set the parameters in the Advanced Options section based on your business requirements. We recommend that you do not modify the settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs. **Note** If you turn on **Enable Plug-in Processing**, specific parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.
Upload Raw Log	If you turn on **Upload Raw Log**, each raw log is uploaded to Log Service as a value of the __raw__ field together with the parsed log.
Topic Generation Mode	The topic generation mode. ○ **Null - Do not generate topic**: This mode is selected by default. In this mode, the topic field is set to an empty string. You do not need to enter a topic to query logs. ○ **Machine Group Topic Attributes**: This mode is used to differentiate logs that are generated by different servers. ○ **File Path RegEx**: In this mode, you must specify a regular expression in the **Custom RegEx** field. The part of a log path that matches the regular expression is used as the topic. This mode is used to differentiate logs that are generated by different users or instances.
Custom RegEx	If you set the Topic Generation Mode parameter to **File Path RegEx**, you must enter a custom regular expression.
Log File Encoding	The encoding format of log files. Valid values: ○ utf8: UTF-8 encoding format ○ gbk: GBK encoding format

Parameter	Description
Timezone	The time zone where logs are collected. Valid values: ○ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ○ Custom: If you select this value, you must select a time zone.
Timeout	The timeout period of log files. If a log file is not updated within the specified period, Logtail reckons the file to be timed out. Valid values: ○ Never: All log files are continuously monitored and never time out. ○ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select **30 Minute Timeout**, you must specify the **Maximum Timeout Directory Depth** parameter. Valid values: 1 to 3.
Filter Configuration	Only logs that **meet all filter conditions** are collected. Examples: ○ Collect logs that meet specified conditions: If you set **Key** to **level** and **Regex** to **WARNING|ERROR**, only WARNING-level and ERROR-level logs are collected. ○ Filter out logs that do not meet specified conditions. ■ If you set **Key** to **level** and **Regex** to **^(?!.*(INFO|DEBUG)).***, INFO-level or DEBUG-level logs are not collected. ■ If you set **Key** to **url** and **Regex** to **.*^(?!.*(healthcheck)).***, logs whose URLs contain healthcheck are not collected. For example, the log is not collected if the value of the Key field is url and the value of the Value field is /inner/healthcheck/jiankong.html.

8. Configure indexes in the Configure Query and Analysis step. Click **Next**.

   By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Enable the index feature and configure indexes for a Logstore.

   > **Note**
   > ○ To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings of Field Search prevail.
   > ○ If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

After you complete the settings, you can start to collect Apache logs.

28.1.3.1.4.9. Configure parsing scripts

This topic describes how to configure log contents for log collection.

Specify a method to separate log lines

A complete access log such as an NGINX access log occupies a line. Separate multiple log entries with line breaks. For example, the following shows two access logs:

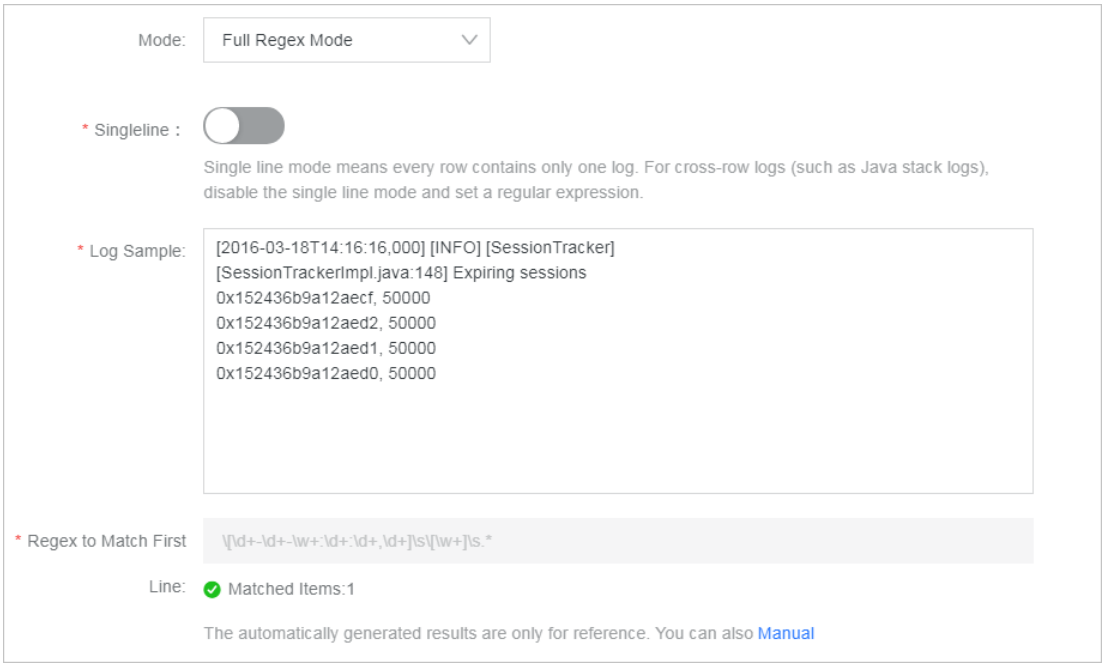
```
10.1.1.1 - - [13/Mar/2016:10:00:10 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)"
10.1.1.1 - - [13/Mar/2016:10:00:11 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)"
```

For Java applications, a log entry usually spans several lines. Therefore, log entries are separated based on the identifier at the beginning of each log entry. The following example shows a Java application log.

```
[2016-03-18T14:16:16,000] [INFO] [SessionTracker] [SessionTrackerImpl.java:148] Expiring sessions
0x152436b9a12aecf, 50000
0x152436b9a12aed2, 50000
0x152436b9a12aed1, 50000
0x152436b9a12aed0, 50000
```

The preceding Java application log entries each start with a time field. The regular expression that matches these time fields is `[\d+-\d+-\w+:\d+:\d+,\d+]\s. .*`. You can enter information in the Log Service console as shown in the following figure.

Full regular expression mode



Extract log fields

To conform to the data models of Log Service, a log contains one or more key-value pairs. If you want to extract specific fields for analysis, you must set a regular expression. If you do not want to process the contents of a log, you can treat the log as a key-value pair.

You can determine whether to extract fields from the preceding NGINX access log.

- Extract fields

  The regular expression is `(\S+)\s-\s-\s\[(\S+)\s[^]]+]\s"(\w+). *`. The extracted fields are `10.1.1.1`, `13/Mar/2016:10:00`, and `GET`.

- Extract all

  The regular expression is `(. *)`. The extracted field is `10.1.1.1 - - [13/Mar/2016:10:00:10 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)"`.

Specify a log time

To conform to the data models of Log Service, a log must have a time field in the Unix timestamp format. You can use the system time when Logtail collects a log or the time in the log contents as the log time.

For the preceding NGINX access log:

- If you extract the time field from the log contents as the log time, the time is `13/Mar/2016:10:00:10` and the time expression is `%d/%b/%Y:%H:%M:%S`.
- If you use the system time when the log was collected as the log time, the log time is converted into a timestamp.

28.1.3.1.4.10. Configure the time format

Each log in Log Service must have a timestamp that records the log generation time. When you collect logs from log files, Logtail must extract the timestamp string of each log and parse it into a timestamp. Therefore, you need to specify a timestamp format to facilitate parsing.

In Linux, Logtail supports all time formats provided by the strftime() function. If a timestamp string can match one of the time formats that are provided by the strftime() function, Logtail can parse and use the timestamp string.

> **Note**
> - The log timestamp is accurate to seconds. Therefore, you need to specify seconds in a time format, without the need for other information such as milliseconds or microseconds.
> - In addition, you need to configure the time field rather than other information.

Common log time formats supported by Logtail

The timestamp strings of logs have diverse formats. To make configuration easier, the following table lists the common log time formats supported by Logtail:

Format	Description	Example
%a	The abbreviation of a day in a week.	Fri
%A	The day in a week.	Friday
%b	The abbreviation of a month.	Jan
%B	The month name.	January
%d	The numerical day in a month. Valid values: 01 to 31.	07 and 31
%h	The abbreviation of a month. The format is equivalent to `%b`.	Jan
%H	The hour in the 24-hour format.	22
%I	The hour in the 12-hour format.	11
%m	The numerical month.	08
%M	The numerical minute. Valid values: 00 to 59.	59
%n	A line break.	Line break

Format	Description	Example
%p	The local time in the a.m. or p.m. format.	AM and PM
%r	The time in the 12-hour format. The format is equivalent to `%I:%M:%S %p`.	11:59:59 AM
%R	The time includes hours and minutes. The format %R is equivalent to `%H:%M`.	23:59
%S	The numerical second. Valid values: 00 to 59.	59
%t	A tab.	Tab
%y	The two-digit numerical year. Valid values: 00 to 99.	04 and 98
%Y	The four-digit numerical year.	2004 and 1998
%C	The numerical century. Valid values: 00 to 99.	16
%e	The numerical day in a month. Valid values: 1 to 31. A single digit is preceded by a space.	7 and 31
%j	The numerical day in a year. Valid values: 001 to 366.	365
%u	The numerical day in a week. Valid values: 1 to 7, in which 1 represents Monday.	2
%U	The numerical week in a year. Sunday is the first day of a week. Valid values: 00 to 53.	23
%V	The numerical week in a year. Monday is the first day of a week. If a week has four or more days that start from January 1, the week is treated as the first week. Otherwise, the next week is treated as the first week. Valid values: 01 to 53.	24
%w	The numerical day in a week. Valid values: 0 to 6, in which 0 represents Sunday.	5
%W	The numerical week in a year. Monday is the first day of a week. Valid values: 00 to 53.	23

Format	Description	Example
%c	The standard date and time.	To specify more information such as the long date and short date, you can use the preceding formats to provide exact expression.
%x	The standard date.	To specify more information such as the long date and short date, you can use the preceding formats to provide exact time expressions.
%X	The standard time.	To specify more information such as the long date and short date, you can use the preceding formats to provide exact expression.
%s	The Unix timestamp.	1476187251

Example

The following table lists the common log time formats, examples, and corresponding time expressions.

Log time format	Example	Time expression
Custom	2017-12-11 15:05:07	%Y-%m-%d %H:%M:%S
Custom	[2017-12-11 15:05:07.012]	[%Y-%m-%d %H:%M:%S]
RFC822	02 Jan 06 15:04 MST	%d %b %y %H:%M
RFC822Z	02 Jan 06 15:04 -0700	%d %b %y %H:%M
RFC850	Monday, 02-Jan-06 15:04:05 MST	%A, %d-%b-%y %H:%M:%S
RFC1123	Mon, 02 Jan 2006 15:04:05 MST	%A, %d-%b-%y %H:%M:%S
RFC3339	2006-01-02T15:04:05Z07:00	%Y-%m-%dT%H:%M:%S
RFC3339Nano	2006-01-02T15:04:05.999999999Z07:00	%Y-%m-%dT%H:%M:%S

28.1.3.1.4.11. Import historical logs

Logtail collects incremental logs by default. If you want to import historical logs, use the historical log importing feature of Logtail.

Prerequisites

To collect logs from Linux servers, use Logtail 0.16.15 or later. To collect logs from Windows servers, use Logtail 1.0.0.1 or later. To ensure successful log collection, update Logtail to the latest version.

Context

Logtail collects log files based on events. The system captures events by detecting or polling files for changes at intervals. Additionally, Logtail can load events from local files to trigger log collection. Logtail implements historical log collection based on these local events.

You can import historical log files from the Logtail installation directory. The location of the directory varies based on the operating system.

- Linux: */usr/local/ilogtail*
- Windows:
  - 32-bit: *C:\Program Files\Alibaba\Logtail*
  - 64-bit: *C:\Program Files (x86)\Alibaba\Logtail*

> **Note**
> - The maximum interval between the time a local event is generated and the time the local event is imported is one minute.
> - Loading local configurations is a special action. Therefore, Logtail sends the `LOAD_LOCAL_EVENT_ALARM` alert to your server to notify you of this action.
> - If you want to import a large number of log files, we recommend that you modify the Logtail startup configuration to increase the upper limit of CPU to 2.0 GHz or more and the upper limit of the memory size to 512 MB or more. For more information, see Set Logtail startup parameters.

Procedure

1. Configure log collection.

   If a collection configuration is only used to import historical log files, you can specify a collection directory that does not exist. For more information, see Configure text log collection.

2. Obtain a unique identifier for a collection configuration.

   Obtain the unique identifier in the *user_log_config.json* file stored in the installation directory of Logtail. In Linux, use the **grep** command in the directory to query the unique identifier. In Windows, use tools such as Notepad to query the unique identifier.

   To query a unique identifier in a Linux operating system, run the following command:

   ```
   grep "##" /usr/local/ilogtail/user_log_config.json | awk '{print $1}'
                       "##1.0##log-config-test$multi"
                       "##1.0##log-config-test$ecs-test"
                       "##1.0##log-config-test$metric_system_test"
                       "##1.0##log-config-test$redis-status"
   ```

3. Add local events.

   Local events are stored in the *local_event.json* file that resides in the installation directory of Logtail. The file is in the JSON format. The syntax is:

   ```
   [
     {
       "config" : "${your_config_unique_id}",
       "dir" : "${your_log_dir}",
       "name" : "${your_log_file_name}"
      },
     {
      ...
      }
      ...
   ]
   ```

   - Parameters

Parameter	Description	Example
config	The unique identifier that is obtained in Step 2.	##1.0##log-config-test$ecs-test

Parameter	Description	Example
dir	The directory where logs are stored. **Note** The directory cannot end with a slash (/).	/data/logs
name	The name of a log file. Wildcards are supported.	For example, access.log.2018-08-08 and access.log*

Note To prevent Logtail from loading invalid JSON files, save local event configurations to a temporary file, edit the configurations in the temporary file, and copy the contents to the *local_event.json* file.

- Configuration examples

  In a Windows system, you can use tools such as Notepad to add local events to the *local_event.json* file. In a Linux system, add local events as follows:

```
$ cat /usr/local/ilogtail/local_event.json
[
  {
    "config": "##1.0##log-config-test$ecs-test",
    "dir": "/data/log/",
    "name": "access.log.
  },
  {
    "config": "##1.0##log-config-test$ecs-test",
    "dir": "/tmp",
    "name": "access.log.2017-08-09"
  }
]
```

What's next

- Check whether Logtail has loaded configurations

  After you save the local_event.json file, Logtail loads the configuration file to the memory within one minute and clears the contents of the local_event.json file.

  To check whether Logtail has read local events, use the following methods:

  - If the contents of the local_event.json file are cleared, it indicates that Logtail has read the local events.
  - Check whether the *ilogtail.LOG* file in the Logtail installation directory contains the process local event keywords. If the contents of the local_event.json are cleared and these keywords cannot be found, the local configuration file may be screened due to invalid contents.

- Check whether the configuration is loaded but no data is collected

  Possible causes are as follows:

  - The configuration is invalid.
  - The local config file does not exist.
  - The log file does not exist in the path specified in the collection configuration.

- The log file has been collected by Logtail.

28.1.3.1.4.12. Generate a topic

This topic describes how to generate a topic in the Log Service console. After you generate a topic, you can use the topic to group logs. You can specify topics for logs when these logs are written. You can use a topic as a filter when you query logs.

Topic generation modes

You can set a topic when you use Logtail to collect logs or when you use API operations or SDKs to upload logs. The following topic generation modes are available in the Log Service console: **Null - Do not generate topic**, **Server Group Topic Attributes**, and **File Path RegEx**.

- **Null - Do not generate topic**

  When you configure Logtail in the Log Service console to collect text logs, the default topic generation mode is **Null - Do not generate topic**. In this mode, no topic is generated and query logs without specifying a topic.

- **Server Group Topic Attributes**

  You can use this mode to identify logs that are generated from multiple servers. Logs from multiple servers can be stored in the same file or directory. To identify these logs based on topics during log collection, you can create server groups and add the servers into different groups. When you create server groups, you must specify a unique **topic attribute** for each server group and set **Topic Generation Mode** to **Server Group Topic Attributes**. After you complete the configuration, apply the Logtail settings to the server groups.

  If the **Server Group Topic Attributes** mode is selected, Logtail uploads the topic attribute of each server group as topics to Log Service. When you query logs, you must specify the topic of the target server group as a filter.

- **File Path RegEx**
  - You can use this mode to differentiate between logs that are generated by multiple users or instances. If Log Service stores logs in different directories for different users or instances, duplicate sub-directory names or log file names may exist in these directories. As a result, Log Service cannot identify the source of logs. You can select **File Path RegEx** in the **Topic Generation Mode** field. Enter a regular expression that matches an absolute file path, and set an instance name as a topic.
  - If you select **File Path RegEx**, Logtail uses an instance name as the topic of the logs that Logtail uploads to Log Service. The topic generated varies based on your directory structure and configuration. You must specify an instance name as a topic when you query logs. For example, the following directory structure includes directories that each store logs generated by different users or instances:

    ```
    /logs
      | - /userA/serviceA
        | - service.log
      | - /userB/serviceA
        | - service.log
      | - /userC/serviceA
        | - service.log
    ```

- If you want to extract multiple separate fields from a file path, use a multi-layer extraction method of `?P<key>`. The value of the key can contain lowercase letters and digits. For example:

  ```
  /home/admin/serviceA/userB/access.log
  \/home\/admin\/(? P<service>[^\/]+)/(? P<user>[^/]+)/. *
  ```

  The following custom tags are created for logs:

  ```
  "__tag__ : service : serviceA"
  "__tag__ : user : userB"
  ```

  > **Note** Logtail 0.16.19 and later are supported.

- If you specify the */logs* file path and the *service.log* file name in a regular expression, Logtail collects logs from the preceding directories that contain the service.log file and uploads the logs to Log Service. However, Log Service cannot identify the log source based on log contents. You can select **File Path RegEx** in the **Topic Generation Mode** field, and enter the `\/(.*)\/serviceA\/.*` regular expression to extract instance names. After the configuration is complete, the following topics are generated for logs in different directories: userA, userB, and userC. You can specify a topic as a filter to query logs.

> **Note** You must escape the forward slashes (`/`) in the file path that the regular expression contains.

- **Static topic generation**

  You can select **File Path RegEx** in the **Topic Generation Mode** field, and enter **customized:// + custom topic** in the `Custom RegEx` field.

  > **Note** Logtail 0.16.21 and later are supported.

Set a log topic

1. Configure Logtail in the Log Service console. For more information, see Configure text log collection.

   To set the topic generation mode to **Server Group Topic Attributes**, go to the **Topic** section on the server group creation or modification page.
2. In the Logtail Configuration for Data Import step, click **Advanced Options** and select a **topic generation mode**.

28.1.3.1.5. Custom plug-ins

Context

Log Service allows you to collect text logs and system logs through Logtail. Logtail supports connections with multiple data sources, such as HTTP or MySQL query results and MySQL binary logs.

You can collect HTTP request data and upload the processing results to Log Service in real time to check service availability check and continuous availability monitoring. You can configure MySQL query results as the data source, and then synchronize incremental data based on custom IDs or time. You can also configure an SQL data source to synchronize MySQL binary logs, subscribe to database changes, and query or analyze logs in real time.

> **Note** This feature is only supported on Linux and must be used together with Logtail. 0.16.0 or later versions. For more information, see Install Logtail in Linux.

Configuration process

1. Configure a method that is used to collect logs from the data source.

   Different Logtail configurations for different data sources. Select a Logtail configuration according to your data source.

2. Configure a processing method.

   Logtail provides multiple processing methods for binary logs, MySQL query results, NGINX monitoring data, and HTTP input sources. You can configure multiple processing methods for a single input source. Each input source supports all processing methods. Logtail runs the configured processing methods in sequence.

   For more information, see Configure data processing methods.

3. Apply the configurations to the machine group.

   After you configure the collection and processing methods, apply them to the specified machine group. Then, Logtail automatically applies the configurations and starts data collection.

28.1.3.1.5.1. Collect MySQL binary logs

Logtail is used as a MySQL slave. It is used to collect binary logs from a MySQL master. Logtail collects binary logs by using a similar method to Alibaba Canal. This improves the efficiency of log collection.

Features

- Allows you to collect incremental data of databases in the form of binary logs to improve performance. Supports MySQL databases such as ApsaraDB RDS for MySQL.
- Supports multiple database filters, such as regular expressions.
- Allows you to set binary log file positions.
- Allows you to record synchronization statuses by using the checkpoint mechanism.

Limits

- MySQL binary logs are available only for Logtail 0.16.0 or later versions that you install on Linux. For more information about how to update Logtail and view Logtail versions, see Install Logtail in Linux.
- Binary logs in the ROW format must be enabled for MySQL databases. By default, binary logs in the ROW format are enabled for RDS instances.

```
# Check whether binary logs are enabled.
mysql> show variables like "log_bin";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| log_bin       | ON    |
+---------------+-------+
1 row in set (0.02 sec)
# View the format of binary logs.
mysql> show variables like "binlog_format";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| binlog_format | ROW   |
+---------------+-------+
1 row in set (0.03 sec)
```

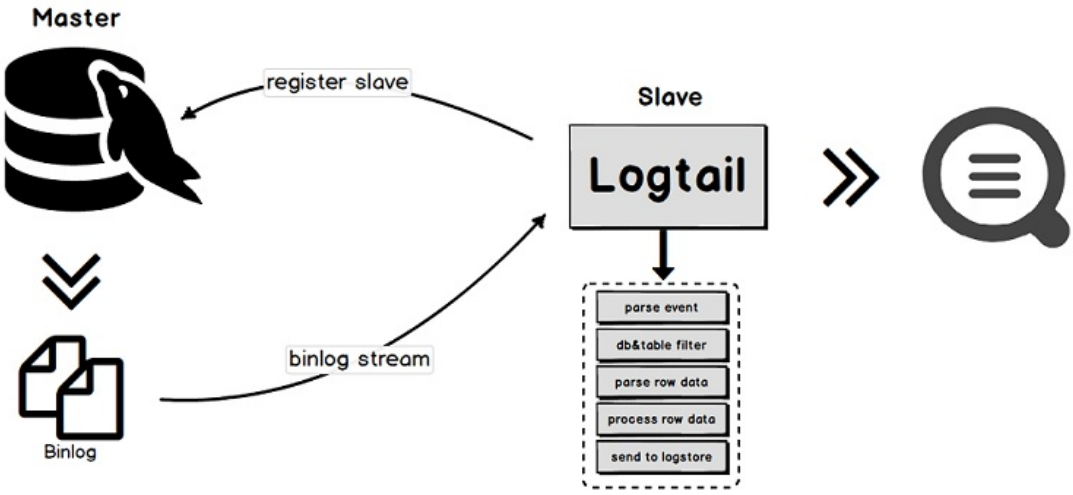
- Each server ID must be unique. Make sure that the ID of each slave to be synchronized is unique.
- Limits for RDS databases:
  - Logtail cannot be installed on an RDS instance. You must install Logtail on an ECS instance that can communicate with the destination RDS instance.

- Secondary RDS databases cannot be used to collect binary logs. You must configure a primary RDS database to collect binary logs.

Implementation

Logtail enables communication between master and slave MySQL servers. The process of how master and slave MySQL servers communicate is provided in the following information:

1. Logtail is used as a MySQL slave. It can also be used to send dump requests to the MySQL master.
2. After the dump requests are received, the MySQL master delivers its binary logs to Logtail in real time.
3. Logtail parses and filters binary logs, and then uploads the results to Log Service.



Scenarios

The MySQL binary logging feature applies to scenarios in which you need to synchronize large amounts of data and meet high performance requirements.

- Query the incremental data of databases in real time.
- Audit operations that are performed on databases.
- Use Log Service to query data, visualize query results, transform data for stream processing, export data to MaxCompute for offline computing, and export log to Object Storage Service (OSS) for long-term storage.

Data reliability

We recommend that you enable the global transaction identifier (GTID) feature of the MySQL server and upgrade Logtail to version 0.16.15 or later. This prevents repeated data collection during a primary/secondary server switchover and ensures data reliability.

- Incomplete data collection: If the network between Logtail and the MySQL server is disconnected for a long period of time, some data may not be collected.

  A MySQL binary log plug-in is used as a MySQL slave to collect binary logs from the master server. Logtail establishes a connection with the master server to obtain data from the server. If the network between Logtail and the master node is disconnected, the master node still generates new binary logs and deletes expired binary logs. After the connection is reestablished and Logtail is reconnected to the master server, Logtail uses the last checkpoint to request binary log data from the master server. However, if the network is disconnected for a long period of time, the data generated after the checkpoint may be deleted. In this case, the recovery mechanism specifies the new point at which Logtail resumes collecting binary logs. The new point is the most recent binary log file position. If the network is disconnected for a long period of time, some data generated between the checkpoint and the new data collection point may not be collected.

- Repeated data collection: If the ordinal numbers of binary logs on the master and slave servers are different and a master/slave switchover occurs, repeated data collection may occur.

If the MySQL master-slave synchronization is configured, the master server synchronizes the generated binary log data to the slave server. Then, the slave server stores the received binary log data to the local binary log file. If the ordinal numbers of binary logs on the master and slave servers are different, a master/slave switchover occurs. In this case, the mechanism that uses a binary log file name and an offset as the checkpoint causes repeated data collection.

For example, assume that a data entry ranges from `(binlog.100, 4)` to `(binlog.105, 4)` on the master server, and ranges from `(binlog.1000, 4)` to `(binlog.1005, 4)` on the slave. Logtail has obtained the data from the master server and updated the checkpoint to `(binlog.105, 4)`. In this case, if a master/slave switchover occurs without exception, Logtail continues to obtain binary logs from the new master server based on the local checkpoint `(binlog.105, 4)`. The new master server returns the data entries that range from `(binlog.1000, 4)` to `(binlog.1005, 4)` to Logtail. This is because the ordinal numbers of these data entries on the new master server are greater than the ordinal numbers of data entries requested by Logtail. As a result, log data is repeatedly collected.

Parameter

The type of input sources is `service_canal`.

Parameter	Type	Required	Description
Host	string	No	The IP address of the host where the database resides. Default value: 127.0.0.1.
Port	int	No	The port number that you can use to connect with the database. Default value: 3306.

User	string	No	The database username. Default value: root.
			The configured user must have the read permissions on the source database and the MySQL REPLICATION permission. Example:
			`CREATE USER canal IDENTIFIED BY 'canal';`
			`GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *. * TO 'canal'@'%';`
			`-- GRANT ALL PRIVILEGES ON *. * TO 'canal'@'%' ;`
			`FLUSH PRIVILEGES;`

Parameter	Type	Required	Description
Password	string	No	The database password. By default, this parameter is unspecified. If you require a high level of data security, we recommend that you set both the username and the password to xxx. After your configurations are synchronized to the on-premises server, find the Password parameter in the */usr/local/ilogtail/user_log_config.json* file and modify the value. **Note** • After you modify the password, use the **sudo /etc/init.d/ilogtaild stop; sudo /etc/init.d/ilogtaild start** command to restart Logtail. • If you modify the value of the Password parameter on the web and synchronize your configurations to the on-premises server, the configurations on the on-premises server are overwritten. You can change the configurations on the on-premises server later.
ServerID	int	No	The ID of a MySQL slave whose role is assumed by Logtail. Default value: 125. **Note** In a MySQL database, each ID must be unique. Otherwise, synchronization fails.
IncludeTables	String array	Yes	The names of matched tables. Each value contains a database name and a table name, for example, `test_db.test_table`. You must specify a regular expression for the parameter. Logtail does not collect incremental data from tables whose names do not match the regular expression. To collect incremental data from all tables of a database, set the value of the IncludeTables parameter to `.*\\..*`. **Note** If an exact match is required, make sure that the regular expression is prefixed by `^`. In this case, you must also make sure that the regular expression is suffixed by `$`. Example: `^test_db\\.test_table$`.

Parameter	Type	Required	Description
ExcludeTables	String array	No	The names of excluded tables expressed as a regular expression. The name of a table must include the name of the database to which the table belongs, for example, `test_db.test_table`. If a table meets one of the conditions specified in the parameter, the table is not collected. If you do not specify this parameter, incremental data from all tables is collected. **Note** If an exact match is required, make sure that the regular expression is prefixed by `^`. In this case, you must also make sure that the regular expression is suffixed by `$`. Example: `^test_db\\.test_table$`.

StartBinName	string	No	The name of the first binary log file that is collected by Logtail. If you do not specify this parameter, Logtail starts to collect binary log files that are generated from the current time.
			To collect data from a specific location, view the name of the current binary log file and the file offset. Then, set `StartBinName` and `StartBinLogPos` to actual values. Example:
			`# Set StartBinName to mysql-bin.000063 and StartBinLogPos to 0.`
			`mysql> show binary logs;`
			`+------------------+-----------+`
			`| Log_name | File_size |`
			`+------------------+-----------+`
			`| mysql-bin.000063 | 241 |`
			`| mysql-bin.000064 | 241 |`
			`| mysql-bin.000065 | 241 |`
			`| mysql-bin.000066 | 10778 |`
			`+------------------+-----------+`
			`4 rows in set (0.02 sec)`
			Note
			If you set the `StartBinName` parameter, a large amount of traffic is generated during the first collection.

StartBinLogPos	int	No	The offset of the first binary log file that is collected. Default value: 0.
EnableGTID	bool	No	Specifies whether to add GTID. Default value: true. If the value is false, no GTID is added to uploaded data.
EnableInsert	bool	No	Specifies whether to collect log events triggered by INSERT operations. Default value: true. If the value is false, INSERT events are not collected.
EnableUpdate	bool	No	Specifies whether to collect UPDATE events. Default value: true. If the value is false, UPDATE events are not collected.

Parameter	Type	Required	Description
EnableDelete	bool	No	Specifies whether to collect DELETE events. Default value: true. If the value is false, DELETE events are not collected.
EnableDDL	bool	No	Specifies whether to collect data definition language (DDL) events. Default value: false. If the value is false, DDL events are not collected. **Note** This parameter does not support the `IncludeTables` or `ExcludeTables` filtering methods.
Charset	string	No	The encoding method. Default value: `utf-8`.
TextToString	bool	No	Specifies whether to convert data of the text type into a string. Default value: false.
PackValues	bool	No	Specifies whether to encapsulate event data into the JSON format. Default value: false. If the value is false, event data is not encapsulated. If this feature is enabled, Logtail encapsulates event data into the data and old_data fields in the JSON format. The old_data field is available only for ROW_UPDATE events. For example, assume that a table has three fields named c1, c2, and c3. If this feature is disabled, the ROW_INSERT event data contains three fields c1, c2, and c3. If this feature is enabled, c1, c2, and c3 are encapsulated into one data field and the value is `{"c1":"...", "c2": "...", "c3": "..."}`. **Note** This parameter is available only for Logtail V0.16.19 and later.
EnableEventMeta	bool	No	Specifies whether to collect event metadata. Default value: false. If the value is false, event metadata is not collected. The metadata of binary log events includes `event_time`, `event_log_position`, `event_size`, and `event_server_id`. **Note** This parameter is available only for Logtail V0.16.21 and later.

Procedure

Synchronize data from tables whose names do not end with `_inner` in the `user_info` RDS database.

1. Log on to the Log Service console.
2. Select a data source.

   Click **Import Data**. On the **Import Data** page, select **MYSQL BinLog**.
3. Select a destination project and Logstore, and then click **Next**.

   You can also click **Create Now** to create a project and a Logstore.

   If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

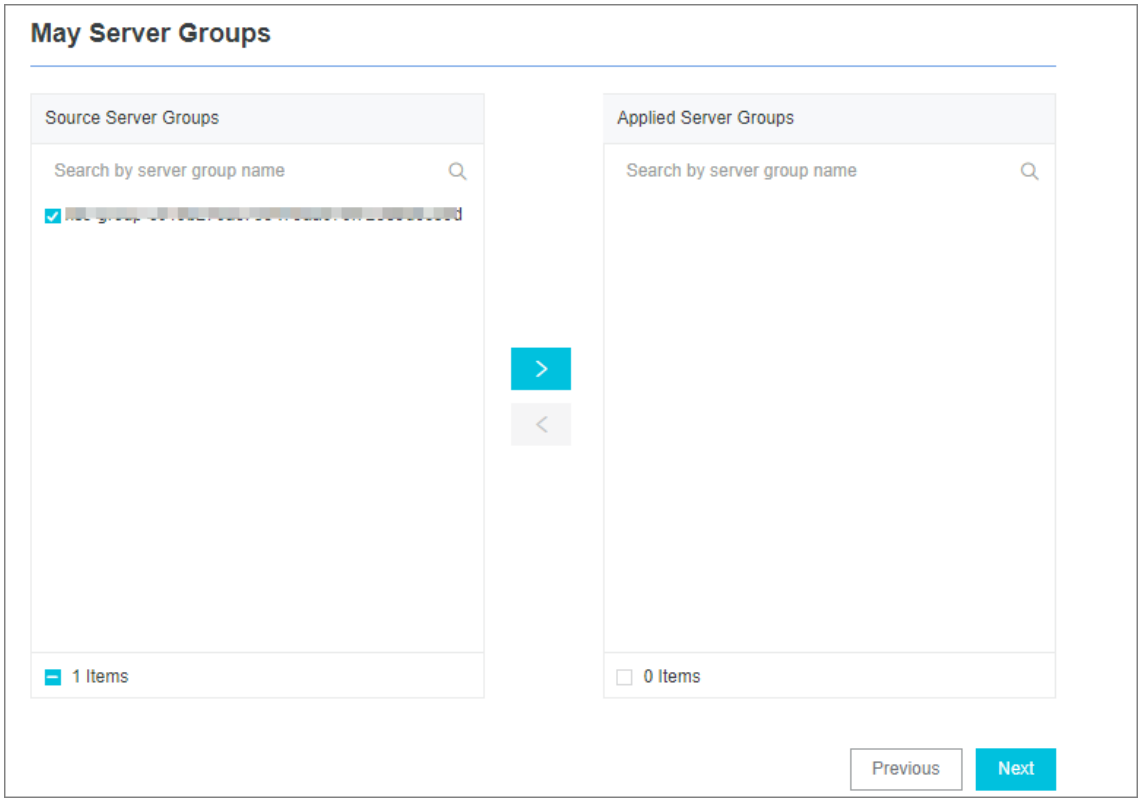
4. Create a machine group and click **Next**.

   Before you can create a machine group, you must install Logtail.

   Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

   After you install Logtail, click **Complete Installation** to create a machine group. If a machine group is created, click **Use Existing Machine Groups** to select the machine group.

5. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



> **Notice** If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

6. Configure the data source.

   Set the Config Name and Plug-in Config fields.

   In the **Plug-in Config** field, modify the parameter settings in the default configuration template based on your business requirements.

```
{
 "inputs": [
     {
         "type": "service_canal",
         "detail": {
             "Host": "************.mysql.rds.aliyuncs.com",
             "Port": 3306,
             "User" : "root",
             "ServerID" : 56321,
             "Password": "*******",
             "IncludeTables": [
                 "user_info\\.. *"
             ],
             "ExcludeTables": [
                 ". *\\. \\S+_inner"
             ],
             "TextToString" : true,
             "EnableDDL" : true
         }
     }
 ]
}
```

- *inputs*: specifies the collection configurations. This parameter is required. You must configure statements to collect data based on your data source.
- *processors*: specifies the processing method. This parameter is optional. For more information about how to set a processing method, see Configure data processing methods.

7. Configure indexes in the Configure Query and Analysis step. Click **Next**.

   By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Enable the index feature and configure indexes for a Logstore.

   > **Note**
   > - To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings of Field Search prevail.
   > - If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

8. (Optional)Modify the configurations on the server.

   If you do not enter the actual URL, username, or password in **Plug-in Config**, you must replace them with actual values after the configurations are synchronized to the server.

   > **Note** If you have entered the actual information, skip this step.

   i. Log on to the server where Logtail is installed, find the `service_canal` keyword in the `/usr/local/ilogtail/user_log_config.json` file, and then set related fields. These fields include `Host`, `User`, and `Password`.

   ii. Run the following command to restart Logtail:

   ```
   sudo /etc/init.d/ilogtaild stop; sudo /etc/init.d/ilogtaild start
   ```

The configurations that are used to collect binary logs are completed. If changes are made to your database, Logtail immediately collects the updated data and uploads the data to Log Service.

> **Note** Logtail collects incremental binary logs. If no data is collected, check whether changes are made to the table in your database after the configurations are updated.

Metadata fields

When you collect binary logs, some metadata is also uploaded. The following table lists the fields of uploaded metadata.

Parameter	Description	Example
`_host_`	The name of the host where the database resides.	`*********.mysql.rds.aliyuncs.com`
`_db_`	The name of the RDS database.	`my-database`
`_table_`	The name of the table.	`my-table`
`_event_`	The type of the event. Valid values:	`row_update`, `row_insert`, and `row_delete`
`_id_`	The ID of the current collection. The value starts from 0 and increments by 1 each time a binary log event is collected.	`1`
`_gtid_`	The GTID.	`7d2ea78d-b631-11e7-8afb-00163e0eef52:536`
`_filename_`	The name of the binary log file.	`binlog.001`
`_offset_`	The offset of the binary log file. The value is updated after each COMMIT operation.	`12876`

Example

After you completed the preceding steps to set a processing method, perform `INSERT`, `UPDATE`, and `DELETE` operations on the `SpecialAlarm` table in the `user_info` database. The following information shows the schema, database operations, and sample logs that are collected by Logtail.

- Schema

```
CREATE TABLE `SpecialAlarm` (
`id` int(11) unsigned NOT NULL AUTO_INCREMENT,
`time` datetime NOT NULL,
`alarmtype` varchar(64) NOT NULL,
`ip` varchar(16) NOT NULL,
`count` int(11) unsigned NOT NULL,
PRIMARY KEY (`id`),
KEY `time` (`time`) USING BTREE,
KEY `alarmtype` (`alarmtype`) USING BTREE
) ENGINE=MyISAM AUTO_INCREMENT=1;
```

- Database operations

  Perform the INSERT, DELETE, and UPDATE operations on the database.

```
insert into specialalarm (`time`, `alarmType`, `ip`, `count`) values(now(), "NO_ALARM", "10.10. **
.***", 55);
delete from specialalarm where id = 4829235  ;
update specialalarm set ip = "10.11. ***.**" where id = "4829234";
```

Create an index for `zc.specialalarm`.

```
ALTER TABLE `zc`.`specialalarm`
ADD INDEX `time_index` (`time` ASC);
```

- Sample logs

  On the data preview or Search & Analysis page, you can view a sample log that corresponds to each operation.

  - INSERT statement

```
__source__:  10.30. **.**
__tag__:__hostname__:  iZbp145dd9fccu*****
__topic__:
_db_:  zc
_event_:  row_insert
_gtid_:  7d2ea78d-b631-11e7-8afb-00163e0eef52:536
_host_:  *********.mysql.rds.aliyuncs.com
_id_:  113
_table_:  specialalarm
alarmtype:  NO_ALARM
count:  55
id:  4829235
ip:  10.10. ***.***
time:  2017-11-01 12:31:41
```

  - DELETE statement

```
__source__:  10.30. **.**
__tag__:__hostname__:  iZbp145dd9fccu****
__topic__:
_db_:  zc
_event_:  row_delete
_gtid_:  7d2ea78d-b631-11e7-8afb-00163e0eef52:537
_host_:  *********.mysql.rds.aliyuncs.com
_id_:  114
_table_:  specialalarm
alarmtype:  NO_ALARM
count:  55
id:  4829235
ip:  10.10. **.***
time:  2017-11-01 12:31:41
```

- UPDATE statement

```
__source__:  10.30. **.**
__tag__:__hostname__:  iZbp145dd9fccu****
__topic__:
_db_:  zc
_event_:  row_update
_gtid_:  7d2ea78d-b631-11e7-8afb-00163e0eef52:538
_host_:  *********.mysql.rds.aliyuncs.com
_id_:  115
_old_alarmtype:  NO_ALARM
_old_count:  55
_old_id:  4829234
_old_ip:  10.10.22.133
_old_time:  2017-10-31 12:04:54
_table_:  specialalarm
alarmtype:  NO_ALARM
count:  55
id:  4829234
ip:  10.11. ***.***
time:  2017-10-31 12:04:54
```

- DDL statement

```
__source__:  10.30. **.**
__tag__:__hostname__:  iZbp145dd9fccu****
__topic__:
_db_:  zc
_event_:  row_update
_gtid_:  7d2ea78d-b631-11e7-8afb-00163e0eef52:539
_host_:  *********.mysql.rds.aliyuncs.com
ErrorCode:  0
ExecutionTime:  0
Query:  ALTER TABLE `zc`.`specialalarm`
ADD INDEX `time_index` (`time` ASC)
StatusVars:
```

Usage notes

We recommend that you increase resource limits on Logtail to process traffic surges and prevent data security risks. If the limits are exceeded, Logtail may be forcibly restarted.

You can modify the resource limits in the */usr/local/ilogtail/ilogtail_config.json* file. Then, you can run the **sudo /etc/init.d/ilogtaild stop;sudo /etc/init.d/ilogtaild start** command to restart Logtail.

The following example shows how to set the CPU limit to two and memory limit to 2,048 MB:

```
{
    ...
    "cpu_usage_limit":2,
    "mem_usage_limit":2048,
    ...
}
```

28.1.3.1.5.2. Collect MySQL query results

This topic describes how to configure Logtail in the Log Service console to collect MySQL query results.

Context

If you want to collect data from a MySQL database, you can install Logtail on a server and connect the server with the database. Then, you can create a Logtail configuration file by executing a custom SQL statement in the Log Service console and deliver the configuration file to Logtail. Logtail can use the custom SQL statement to collect data from the database at regular intervals.

> **Note** This feature applies only to Logtail 0.16.0 and later versions that run on Linux. For more information, see Install Logtail in Linux.

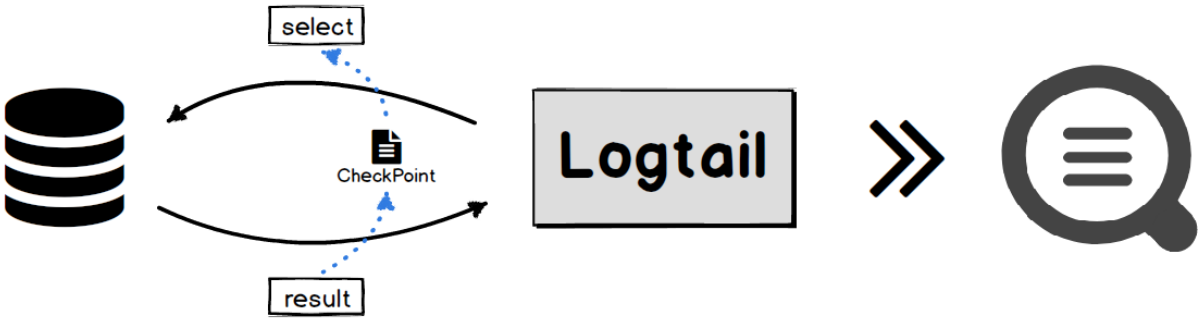
Benefits

Supports MySQL databases such as ApsaraDB RDS for MySQL. When you collect data, you can perform the following operations:

- Paginate query results.
- Set time zones.
- Set timeout periods.
- Store checkpoints.
- Transmit data over the SSL protocol.
- Set the maximum size of data that can be collected at a time.

Implementation

Logtail executes the specified SELECT statement at a regular interval based on the Logtail configurations, and then uploads the query results to Log Service.



When Logtail obtains a query result, Logtail saves the value of the CheckPoint field in the on-premises server. The next time Logtail executes the SELECT statement, Logtail adds the saved value of the CheckPoint field to the SELECT statement. This way, Logtail collects the incremental data of MySQL databases.

Scenarios

- Collect incremental data based on specific marks such as an auto-increment ID or a point in time.
- Customize data synchronization based on specified filtering conditions.

Parameters

The following table describes Logtail parameters. The type of the data source is `service_mysql`.

Parameter	Type	Required	Description
Address	string	No	The address of the MySQL database. Default value: 127.0.0.1:3306.

Parameter	Type	Required	Description
User	string	No	The username of the MySQL database. Default value: root.
Password	string	No	The password of the MySQL database. By default, this parameter is unspecified.
DialTimeOutMs	int	No	The timeout period for the database connection. Unit: milliseconds. Default value: 5000.
ReadTimeOutMs	int	No	The timeout period for data reading. Unit: milliseconds. Default value: 5000.
StateMent	string	Yes	The SQL statement.
Limit	bool	No	Specifies whether to paginate query results by using the LIMIT clause. Default value: false.
PageSize	int	No	The number of log entries to return on each page. You must specify this parameter if you set the Limit parameter to true.
MaxSyncSize	int	No	The maximum number of log entries that are synchronized at a time. Default value: 0. This value indicates that no limit is set for the size of data that can be synchronized at a time.
CheckPoint	bool	No	Specifies whether to use checkpoints during data collection. Default value: false.
CheckPointColumn	string	No	The name of the checkpoint column. You must specify this parameter if you set the CheckPoint parameter to true.
CheckPointColumnType	string	No	The type of the checkpoint column. Valid values: `int` and `time`.

Parameter	Type	Required	Description
CheckPointStart	string	No	The initial value of the checkpoint.
CheckPointSavePerPage	bool	No	If this parameter is set to true, a checkpoint is saved each time query results are paginated. If this parameter is set to false, a checkpoint is saved after each synchronization.
IntervalMs	int	Yes	The synchronization interval. Unit: milliseconds.

Limits

- We recommend that you paginate query results by specifying the `Limit` parameter. If you set the `Limit` parameter to true, the LIMIT clause is automatically appended to the SQL statement specified by the `StateMent` parameter when you run a query.
- If you set the `CheckPoint` parameter to true, the data that is selected based on the `StateMent` parameter must contain the checkpoint column. In addition, the WHERE clause must contain the checkpoint field. The value of the checkpoint field is a question mark (`?`).

  For example, assume that the checkpoint is "id" and the value of the `StateMent` parameter is `SELECT * from ... where id > ?` .
- If you set the `CheckPoint` parameter to true, you must specify the `CheckPointColumn` , `CheckPointColumnType` , and `CheckPointStart` parameters.
- The value of `CheckPointColumnType` can only be set to `int` or `time` . If the value is set to `int` , the int64 data type is used for internal storage. If the value is set to `time` , MySQL DATE, DATETIME, and TIME are supported.

Procedure

The following procedure describes how to synchronize incremental data from a MySQL database to Log Service. In this procedure, the logtail.VersionOs field is synchronized every 10 seconds. The value of the count parameter in this field is greater than 0. The value of the initial checkpoint is 2017-09-25 11:00:00. Log entries are paginated and each page contains 100 log entries. The checkpoint of each page is saved.

1. Log on to the Log Service console.
2. Select a data source.

   Click **Import Data**. On the **Import Data** page, select **MYSQL Query Result-Plug-in**.
3. Select a destination project and Logstore, and then click **Next**.

   You can also click **Create Now** to create a project and a Logstore.

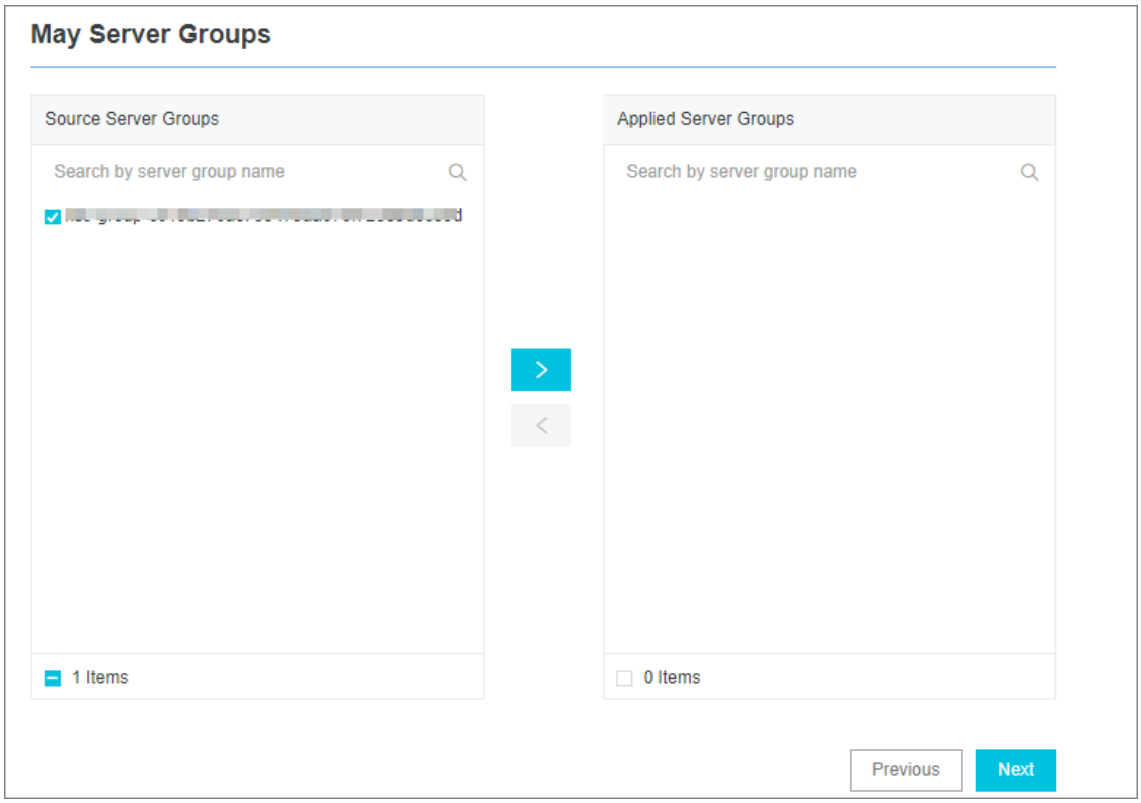
   If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.
4. Create a machine group and click **Next**.

   Before you can create a machine group, you must install Logtail.

   Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

   After you install Logtail, click **Complete Installation** to create a machine group. If a machine group is created, click **Use Existing Machine Groups** to select the machine group.

5. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



> **Notice** If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

6. Configure the data source.
   - In the **Plug-in Config** field, modify the parameter settings in the default configuration template based on your business requirements.
   - `inputs`: specifies the collection configurations. This parameter is required. `processors`: specifies the processing method. This parameter is optional. You must configure statements to collect data based on your data source. You can specify one or more processing methods. For more information, see Configure data processing methods.

> **Note** If you require a high level of data security, we recommend that you set both the username and the password to xxx. After your configurations are synchronized to the on-premises server, find the Password parameter in the /usr/local/ilogtail/user_log_config.json file and modify the value.

The following example shows the configurations:

```
{
 "inputs": [
     {
         "type": "service_mysql",
         "detail": {
             "Address": "********:3306",
             "User": "logtail",
             "Password": "*******",
             "DataBase": "logtail",
             "Limit": true,
             "PageSize": 100,
             "StateMent": "SELECT * from logtail.VersionOs where time > ?",
             "CheckPoint": true,
             "CheckPointColumn": "time",
             "CheckPointStart": "2017-09-25 11:00:00",
             "CheckPointSavePerPage": true,
             "CheckPointColumnType": "time",
             "IntervalMs": 10000
         }
     }
 ]
}
```

7. Configure indexes in the Configure Query and Analysis step. Click **Next**.

   By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Enable the index feature and configure indexes for a Logstore.

   > **Note**
   > - To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings of Field Search prevail.
   > - If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

8. (Optional)Modify the configurations on the server.

   If you do not enter the actual URL, username, or password in the **Specify Data Source** page, you must replace them with actual values after the configurations are synchronized to the server.

   > **Note** If you have entered the actual information, skip this step.

   i. Log on to the server where Logtail is installed, find the `service_mysql` keyword in the `/usr/local/ilogtail/user_log_config.json` file, and then set related fields. These fields include: `Address`, `User`, and `Password`.

   ii. Run the following command to restart Logtail:

   ```
   sudo /etc/init.d/ilogtaild stop; sudo /etc/init.d/ilogtaild start
   ```

Example

After you configure the processing method, you can view the processed data in the Log Service console. The following information shows the schema and sample logs that are collected by Logtail.

- Schema

```
CREATE TABLE `VersionOs` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT COMMENT 'id',
  `time` datetime NOT NULL,
  `version` varchar(10) NOT NULL DEFAULT '',
  `os` varchar(10) NOT NULL,
  `count` int(11) unsigned NOT NULL,
  PRIMARY KEY (`id`),
  KEY `timeindex` (`time`)
)
```

- Sample output

```
"count":  "4"
"id:  "721097"
"os:  "Windows"
"time:  "2017-08-25 13:00:00"
"version":  "1.3.0"
```

28.1.3.1.5.3. Collect syslogs

This topic describes how to use the syslog plug-in of Logtail to collect syslogs from a server.

Prerequisites

Logtail 0.16.13 or a later version is installed on the server.

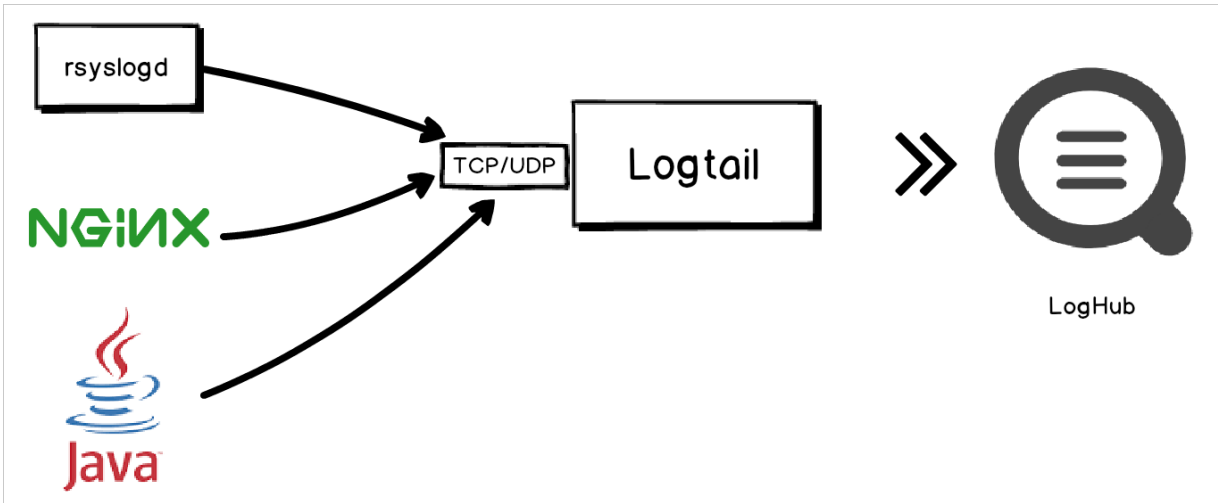
Overview

On a Linux server, local syslogs can be forwarded to the IP address and port of a specified server by using syslog agents such as rsyslog. After you create a Logtail configuration for the specified server, the syslog plug-in of Logtail receives the syslogs over TCP or UDP. In addition, the syslog plug-in parses the received syslogs and extract log fields such as facility, tag (program), severity, and content based on the specified syslog protocol. The syslog protocol can be RFC 3164 or RFC 5424.

> **Note**
> - Logtail installed on a Windows server does not support the syslog plug-in.
> - You can configure multiple syslog plug-ins for Logtail. For example, you can use both TCP and UDP to listen on 127.0.0.1:9999.

Implementation

After the syslog plug-ins start to listen on a specified IP address and port, Logtail can act as a syslog server to collect syslogs from various data sources. These syslogs include system logs collected by rsyslog, access or error logs forwarded by NGINX, and logs forwarded by syslog clients in languages such as Java.



Logtail parameters

The following table describes Logtail parameters. The type of the input is `service_syslog`.

Parameter	Type	Required	Description
Address	String	No	The protocol, address, and port on which the syslog plug-in listens. The syslog plug-in obtains logs based on the value of this parameter. Format: `[tcp/udp]://[ip]:[port]`. Default value: `tcp://127.0.0.1:9999`. **Note** • The specified protocol, address, and port must be the same as those specified for the forwarding rule in the rsyslog configuration file. • If the server on which Logtail is installed has multiple IP addresses, you can set the IP address to 0.0.0.0. It means that the syslog plug-in listens on all IP addresses of the server.
ParseProtocol	String	No	The protocol that is used to parse logs. This parameter is not specified by default, indicating that logs are not parsed. Valid values: • rfc3164: The RFC 3164 protocol is used to parse logs. • rfc5424: The RFC 5424 protocol is used to parse logs. • auto: The syslog plug-in selects a protocol based on the log content.

Parameter	Type	Required	Description
IgnoreParseFailure	Boolean	No	Specifies whether to ignore a parsing failure. Default value: true. Valid values: • true: Logs that fail to be parsed are included in the returned content field. • false: Logs that fail to be parsed are dropped.

Default fields

Field	Type	Description
hostname	String	The hostname. If a hostname is not provided in the log, the hostname of the current host is obtained.
program	String	The tag field in the protocol.
priority	String	The priority field in the protocol.
facility	String	The facility field in the protocol.
severity	String	The severity field in the protocol.
unixtimestamp	String	The timestamp of the log.
content	String	The log content. If the log fails to be parsed, this field contains the complete content of the log.
_ip_	String	The IP address of the current host.

Configure the plug-in of Logtail to collect syslogs

1. Add a forwarding rule for rsyslog.

   Modify the */etc/rsyslog.conf* rsyslog configuration file on the server from which syslogs are collected. Add a forwarding rule at the end of the configuration file. Then, rsyslog forwards syslogs to the specified IP address and port.

   - If you want to collect syslogs of the server by using Logtail on this server, set the forwarding address to 127.0.0.1 and the port to an idle port.
   - If you want to collect syslogs of the server by using Logtail on a second server (Server B), set the forwarding address to the public IP address of the second server and port to an idle port.

   For example, the following forwarding rule indicates that logs are forwarded to 127.0.0.1:9000 over TCP.

   ```
   *. * @@127.0.0.1:9000
   ```

2. Run the following command to restart rsyslog and validate the log forwarding rule:

   ```
   sudo service rsyslog restart
   ```

3. Log on to the Log Service console.
4. Select the data source **Custom Data Plug-in**.

   You can use one of the following three methods to select a data source:

   - On the homepage of the Log Service console, select a data source in the **Import Data** section.
   - In the **Projects** section, click a project name. On the **Overview** page, click **Import Data**, and then select a data source.

- On the Logstores tab in the left-side navigation pane, find a Logstore and click the closing angle bracket (>) in front of the Logstore name. Click the plus sign (+) next to **Data Import**, and then select a data source.

5. Select a destination project and Logstore, and then click **Next**.

   You can also click **Create Now** to create a project and a Logstore.

   If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

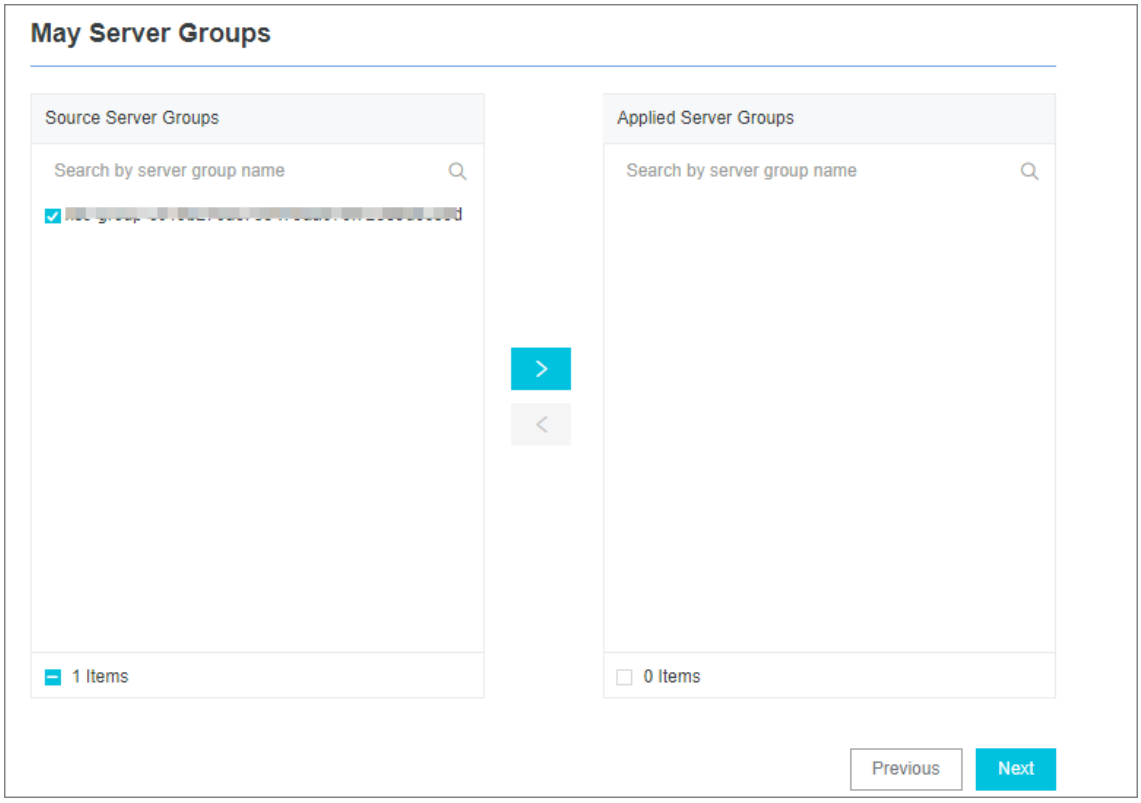
6. Create a machine group and click **Next**.

   Before you can create a machine group, you must install Logtail.

   Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

   After you install Logtail, click **Complete Installation** to create a machine group. If a machine group is created, click **Use Existing Machine Groups** to select the machine group.

7. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



> **Notice** If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

8. Configure the data source.

   Set **Config Name** and **Plug-in Config**.

   The `inputs` section is required. It specifies the collection configuration. The `processors` section is optional. It specifies the processing configuration. You must specify a collection statement for the collection configuration based on the data source. You can specify one or more processing methods for the processing configuration. For more information, see Configure data processing methods.

   The following sample code shows how to use UDP and TCP to listen on 127.0.0.1:9000:

```
{
     "inputs": [
         {
             "type": "service_syslog",
             "detail": {
                 "Address": "tcp://127.0.0.1:9000",
                 "ParseProtocol": "rfc3164"
             }
         },
         {
             "type": "service_syslog",
             "detail": {
                 "Address": "udp://127.0.0.1:9001",
                 "ParseProtocol": "rfc3164"
             }
         }
     ]
 }
```

9. Configure indexes in the Configure Query and Analysis step. Click **Next**.

   By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Enable the index feature and configure indexes for a Logstore.

   > **Note**
   > - To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings of Field Search prevail.
   > - If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

Configure the plug-in of Logtail to collect NGINX logs

NGINX access logs can be forwarded to specified addresses and ports over the syslog protocol. To deliver NGINX access logs as syslogs from a server to Log Service, you can create a Logtail configuration and apply it to the server group to which the server belongs.

1. Add a forwarding rule to the *nginx.conf* configuration file on the NGINX server.

   For example, add the following content to the configuration file.

   ```
   http {
       ...
       # Add this line.
       access_log syslog:server=127.0.0.1:9000,facility=local7,tag=nginx,severity=info combined;
       ...
   }
   ```

2. Run the following command to restart the NGINX service and validate the configuration.

   ```
   sudo service nginx restart
   ```

3. Create a Logtail configuration and apply it to the server group to which the server belongs.

   For more information, see Configure the plug-in of Logtail to collect syslogs.

4. Check whether the Logtail configuration takes effect.

   Run the `curl http://127.0.0.1/test.html` command in shell to generate an access log. If the Logtail configuration takes effect, you can view the log information on the query page of the Log Service console.

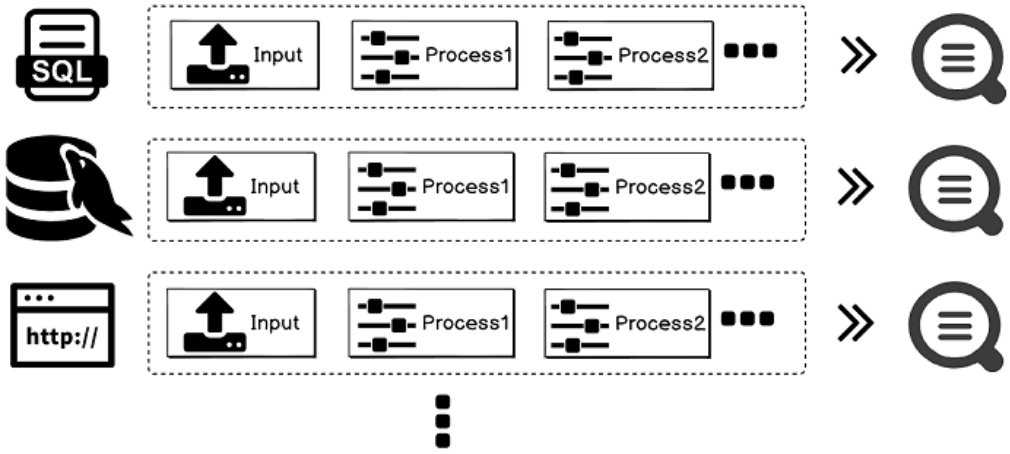
28.1.3.1.5.4. Configure data processing methods

Logtail allows you to configure plug-ins for data processing. Each plug-in defines a processing method. You can configure one or more processing methods for a data source. Then, Logtail executes the processing methods in sequence. This topic describes the available data processing methods and describes how to configure the methods.

Implementation

The following figure shows how collected data is processed.

Implementation



Plug-in elements

When you configure the data processing methods, you must set the key parameter to `processors` and the value parameter to an array of JSON objects. Each object contains the details of a processing method.

Each object contains the `type` and `detail` fields. The `type` field specifies the type of a processing method. The `detail` field contains configuration details.

```
"processors" : [
        {
            "type" : "processor_anchor",
            "detail" : {
                ...
            }
        },
        {
            "type" : "processor_regex",
            "detail" : {
                ...
            }
        }
    ]
```

Processing methods

The following processing methods are supported:

- Extract log fields by using a regular expression
- Extract log fields by using start and stop keywords
- Extract log fields by using a single-character delimiter

- Extract log fields by using a multi-character delimiter
- Convert an IP address into a geographical location
- Filter log fields by using a regular expression
- Add log fields
- Remove log fileds
- Extract log time (Go)
- Expand log fields (JSON)
- Combine log fields (JSON)
- Rename fields
- Extract log time (Strptime)

You can also create a custom method that includes several of the preceding methods. For more information, see Custom methods.

Extract log fields by using a regular expression

This method extracts the fields that match a specified regular expression.

The type of the plug-in is `processor_regex`.

Parameters

Parameter	Type	Required	Description
SourceKey	string	Yes	The name of the field to extract by using the regular expression.
Regex	string	Yes	The regular expression. Enclose the value of fields to extract with parentheses `()`.
Keys	String array	Yes	The array of fields to extract, for example, ["key1", "key2" ...].
NoKeyError	bool	No	Specifies whether to report an error if no field matches the regular expression. Default value: false.
NoMatchError	bool	No	Specifies whether to report an error if the specified regular expression does not match logs. Default value: false.
KeepSource	bool	No	Specifies whether to return the SourceKey parameter. Default value: false. This value specifies that the SourceKey parameter is not returned.
FullMatch	bool	Yes	Specifies whether to extract fields that exactly match the `Regex` parameter. Default value: true. This value specifies that fields that partially match the Regex parameter are extracted.

The following example shows how to extract fields from an access log.

- Input data

```
"content" : "10.200. **. ** - - [10/Aug/2017:14:57:51 +0800] \"POST /PutData?
Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3
A30%20GMT&Topic=raw&Signature=<yourSignature> HTTP/1.1\" 0.024 18204 200 37 \"-\" \"aliyun-sdk-jav
a"
```

- Configurations

```
{
    "type" : "processor_regex",
    "detail" : {"SourceKey" : "content",
         "Regex" : "([\\d\\.] +) \\S+ \\S+ \\[(\\S+) \\S+\\] \"(\\w+) ([^\\\"]*)\" ([\\d\\.]+) (\\
d+) (\\d+) (\\d+|-) \"([^\\\"]*)\" \"([^\\\"]*)\" (\\d+)",
         "Keys"   : ["ip", "time", "method", "url", "request_time", "request_length", "status", "l
ength", "ref_url", "browser"],
         "NoKeyError" : true,
         "NoMatchError" : true,
         "KeepSource" : false
    }
}
```

- Output data

```
"ip" : "10.200. **.**"
"time" : "10/Aug/2017:14:57:51"
"method" : "POST"
"url" : "/PutData? Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun
%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=<yourSignature>"
"request_time" : "0.024"
"request_length" : "18204"
"status" : "200"
"length" : "27"
"ref_url" : "-"
"browser" : "aliyun-sdk-java"
```

Extract log fields by using start and stop keywords

This method uses the `start` and `stop` keywords to extract fields. You can extract fields from the JSON string between the start and stop keywords. You can also expand the JSON string into other forms.

The type of the plug-in is `processor_anchor`.

Parameters

Parameter	Type	Required	Description
SourceKey	string	Yes	The name of the field to extract.
Anchors	Array	Yes	The array of anchors. For more information, see the following table.
NoAnchorError	bool	No	Specifies whether to report an error if no specified keyword is found. Default value: false.
NoKeyError	bool	No	Specifies whether to report an error if no match exists. Default value: false.

Parameter	Type	Required	Description
KeepSource	bool	No	Specifies whether to return the SourceKey parameter. Default value: false. This value specifies that the SourceKey parameter is not returned.

Anchor fields

Parameter	Type	Required	Description
Start	string	Yes	The start keyword. If you do not specify the parameter, Logtail matches the first character of a string.
Stop	string	Yes	The stop keyword. If you do not specify the parameter, Logtail matches the last character of a string.
FieldName	string	Yes	The name of the field to extract.
FieldType	string	Yes	The type of the field to extract. Valid values: `string` and `json`.
ExpondJson	bool	No	Specifies whether to expand JSON strings. Default value: false. This parameter is available only if you specify `json` for the `FieldType` parameter.
ExpondConnecter	string	No	The connector that combines separate field names into a string. Default value: `_`.
MaxExpondDepth	int	No	The maximum depth of JSON expansion. Default value: `0`. This indicates no limit.

The following example shows how to use this method to process input data of multiple types.

- Input data

```
"content" : "time:2017.09.12 20:55:36\tjson:{\"key1\" : \"xx\", \"key2\": false, \"key3\":123.456,
\"key4\" : { \"inner1\" : 1, \"inner2\" : false}}"
```

- Configurations

```
{
   "type" : "processor_anchor",
   "detail" : {"SourceKey" : "content",
      "Anchors" : [
          {
              "Start" : "time",
              "Stop" : "\t",
              "FieldName" : "time",
              "FieldType" : "string",
              "ExpondJson" : false
          },
          {
              "Start" : "json:",
              "Stop" : "",
              "FieldName" : "val",
              "FieldType" : "json",
              "ExpondJson" : true
          }
      ]
  }
}
```

- Output data

```
"time" : "2017.09.12 20:55:36"
"val_key1" : "xx"
"val_key2" : "false"
"val_key3" : "123.456"
"value_key4_inner1" : "1"
"value_key4_inner2" : "false"
```

Extract log fields by using a single-character delimiter

This method uses `single-character delimiters` to split logs into several fields. You can enclose delimited fields by using characters that you specify in the Quote parameter.

The type of the plug-in is `processor_split_char`.

Parameters

Parameter	Type	Required	Description
SourceKey	string	Yes	The name of the field to extract.
SplitSep	string	Yes	The single-character delimiter. You must specify a single character as a delimiter. You can specify a non-printable character such as `\u0001` as a delimiter.
SplitKeys	String array	Yes	The names of the fields that are split, for example, ["key1", "key2"...].
QuoteFlag	bool	No	Specifies whether to use the `Quote` parameter. Default value: false.
Quote	string	No	You must specify a single character. The parameter is available only if you specify true for the `QuoteFlag` parameter. You can specify non-printable characters, such as `\u0001`.

Parameter	Type	Required	Description
NoKeyError	bool	No	Specifies whether to report an error if no match exists. Default value: false.
NoMatchError	bool	No	Specifies whether to report an error if the specified delimiter is not found. Default value: false.
KeepSource	bool	No	Specifies whether to return the SourceKey parameter. Default value: false. This value specifies that the SourceKey parameter is not returned.

The following example shows how to use this method to process logs.

- Input data

```
"content" : "10. **. **. **|10/Aug/2017:14:57:51 +0800|POST|PutData?
Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3
A30%20GMT&Topic=raw&Signature=<yourSignature>|0.024|18204|200|37|-|
aliyun-sdk-java"
```

- Configurations

```
{
   "type" : "processor_split_char",
   "detail" : {"SourceKey" : "content",
      "SplitSep" : "|",
      "SplitKeys" : ["ip", "time", "method", "url", "request_time", "request_length", "status", "l
ength", "ref_url", "browser"]
  }
}
```

- Output data

```
"ip" : "10. **. **.**"
"time" : "10/Aug/2017:14:57:51 +0800"
"method" : "POST"
"url" : "/PutData? Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun
%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=<yourSignature>"
"request_time" : "0.024"
"request_length" : "18204"
"status" : "200"
"length" : "27"
"ref_url" : "-"
"browser" : "aliyun-sdk-java"
```

Extract log fields by using a multi-character delimiter

Similar to the single-character delimiter method, this method uses multi-character delimiters to split logs into several fields. The Quote parameter is unavailable for this method.

The type of the plug-in is `processor_split_string`.

Parameters

Parameter	Type	Required	Description
SourceKey	string	Yes	The name of the field to extract.

Parameter	Type	Required	Description
SplitSep	string	Yes	The multi-character delimiter. You can specify non-printable characters such as `\u0001\u0002`.
SplitKeys	String array	Yes	The names of the fields that are split, for example, ["key1", "key2"...].
PreserveOthers	bool	No	Specifies whether to retain excess fields when the number of split fields exceeds the number of fields defined in the `SplitKeys` parameter. Default value: false.
ExpandOthers	bool	No	Specifies whether to parse excess fields. Default value: false.
ExpandKeyPrefix	string	No	The prefix of the names of excess fields. For example, if you specify `expand_` for the parameter, excess fields are named `expand_1` and `expand_2`.
NoKeyError	bool	No	Specifies whether to report an error if no match exists. Default value: false.
NoMatchError	bool	No	Specifies whether to report an error if no multi-character delimiter is found. Default value: false.
KeepSource	bool	No	Specifies whether to return the SourceKey parameter. Default value: false. The value false specifies that the SourceKey parameter is not returned.

The following example shows how to use this method to process logs.

- Input data

```
"content" : "10. **. **. **|#|10/Aug/2017:14:57:51 +0800|#|POST|#|PutData?
Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3
A30%20GMT&Topic=raw&Signature=<yourSignature>|#|0.024|#|18204|#|200|#|37|#|-|#|
aliyun-sdk-java"
```

- Configurations

```
{
   "type" : "processor_split_string",
   "detail" : {"SourceKey" : "content",
      "SplitSep" : "|#|",
      "SplitKeys" : ["ip", "time", "method", "url", "request_time", "request_length", "status"],
      "PreserveOthers" : true,
      "ExpandOthers" : true,
      "ExpandKeyPrefix" : "expand_"
   }
}
```

- Output data

```
"ip" : "10. **. **.**"
"time" : "10/Aug/2017:14:57:51 +0800"
"method" : "POST"
"url" : "/PutData? Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun
%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=<yourSignature>"
"request_time" : "0.024"
"request_length" : "18204"
"status" : "200"
"expand_1" : "27"
"expand_2" : "-"
"expand_3" : "aliyun-sdk-java"
```

Convert an IP address into a geographical location

This method converts IP address into a geographical location, such as a country, province, city, or geographical coordinates.

Note By default, the Logtail installation package does not include GeoIP databases. You must download these databases to your local host and configure the required parameters. We recommend that you download a version of the GeoIP database that includes the `City` database.

The type of the plug-in is `processor_geoip`.

Parameters

Parameter	Type	Required	Description
SourceKey	string	Yes	The name of the field whose IP address you want to convert into a geographical location.
DBPath	string	Yes	The absolute path of the GeoIP database. The database format is `MMDB`, for example, `/user/data/GeoLite2-City_20180102/GeoLite2-City.mmdb`.
NoKeyError	bool	No	Specifies whether to report an error if no match exists. Default value: false.
NoMatchError	bool	No	Specifies whether to report an error if the specified IP address is invalid or does not match the IP addresses stored in the library. Default value: false.
KeepSource	bool	No	Specifies whether to return the SourceKey parameter. Default value: true.
Language	string	No	The language of the GeoIP database. Default value: `zh-CN`. Make sure that your GeoIP database can be displayed in a language that is suitable for your business.

The following example shows how to use this method to convert an IP address into a geographical location.

- Input data

```
"source_ip" : "**. **. **. **"
```

Download a GeoIP database and install the database on the host where Logtail resides. We recommend that you download MaxMind GeoLite2 that includes the city database.

> **Note** Make sure that the database format is MMDB.

- Configurations

```
{
   "type": "processor_geoip",
    "detail": {
         "SourceKey": "ip",
         "NoKeyError": true,
         "NoMatchError": true,
         "KeepSource": true,
         "DBPath" : "/user/local/data/GeoLite2-City_20180102/GeoLite2-City.mmdb"
    }
}
```

- Output result

```
"source_ip_city_" : "**. **. **.**"
"source_ip_province_" : "Zhejiang"
"source_ip_city_" : "Hangzhou"
"source_ip_province_code_" : "ZJ"
"source_ip_country_code_" : "CN"
"source_ip_longitude_" : "120.********"
"source_ip_latitude_" : "30. ********"
```

Filter log fields by using a regular expression

This method uses regular expressions to filter logs. You can specify conditions in the `Include` and `Exclude` parameters.

The type of the plug-in is `processor_filter_regex`.

Parameters

Parameter	Type	Required	Description
Include	The JSON object that includes key-value pairs.	No	Each key includes a log field. Each value includes a regular expression. If the field value matches the regular expression, the log is collected.
Exclude	The JSON object that includes key-value pairs.	No	Each key includes a log field. Each value includes a regular expression. If the field value matches the regular expression, the log is dropped.

> **Note** A log is collected only if the field value matches the regular expression specified in the `Include` parameter and does not match the regular expression specified in the `Exclude` parameter. Otherwise, the log is dropped.

The following example shows how to use this method to process logs.

- Input data

- Log 1

```
"ip" : "10. **. **.**"
"method" : "POST"
...
"browser" : "aliyun-sdk-java"
```

- Log 2

```
"ip" : "10. **. **.**"
"method" : "POST"
...
"browser" : "chrome"
```

- Log 3

```
"ip" : "192.168. *.*"
"method" : "POST"
...
"browser" : "ali-sls-ilogtail"
```

- Configurations

```
{
   "type" : "processor_filter_regex",
    "detail" : {
         "Include" : {
            "ip" : "10\\..*",
            "method" : "POST"
         },
         "Exclude" : {
            "browser" : "aliyun. *"
         }
    }
}
```

- Output data

Log	Matched	Reason
Log 1	No	The match failed because the value of the browser field matches the regular expression specified in the Exclude parameter.
Log 2	Yes	-
Log 3	No	The match failed because the value of the ip field does not match the regular expression specified in the `Include` parameter. The regular expression matches `IP` addresses that start with `10`.

Add log fields

You can use this method to add multiple fields to a log.

The type of the plug-in is `processor_add_fields`.

> **Note** This method is available for Logtail 0.16.28 or later.

Parameters

Parameter	Type	Required	Description
Fields	map	No	The JSON object that includes key-value pairs. You can specify multiple key-value pairs in the parameter.
IgnoreIfExist	bool	No	Specifies whether to ignore key-value pairs that have the same keys. Default value: false.

The following example shows how to use this method to add fields to logs.

- Input data

```
"aaa1":"value1"
```

- Configurations

```
{
  "processors":[
    {
      "type":"processor_add_fields",
      "detail": {
        "Fields": {
          "aaa2": "value2",
          "aaa3": "value3"
        }
      }
    }
  ]
}
```

- Output data

```
"aaa1":"value1"
"aaa2":"value2"
"aaa3":"value3"
```

Remove log fileds

You can use this method to remove specific fields from logs.

The type of the plug-in is `processor_drop`.

> **Note** The method is available for Logtail 0.16.28 or later.

Parameters

Parameter	Type	Required	Description
DropKeys	array	No	The array that includes a set of strings. You can remove multiple fields from a log.

The following example shows how to remove the `aaa1` and `aaa2` fields from a log.

- Input data

```
"aaa1":"value1"
"aaa2":"value2"
"aaa3":"value3"
```

- Configurations

```
{
  "processors":[
    {
      "type":"processor_drop",
      "detail": {
        "DropKeys": ["aaa1","aaa2"]
      }
    }
  ]
}
```

- Output data

```
"aaa3":"value3"
```

Extract log time (Go)

You can use this method to extract time from a log field and convert the time format.

The type of the plug-in is `processor_gotime`.

> **Note** This plug-in is available for Logtail 0.16.28 or later.

Parameters

Parameter	Type	Required	Description
SourceKey	string	Yes	The name of the log field from which you want to extract time.
SourceFormat	string	Yes	The time that is parsed in Go.
SourceLocation	int	Yes	The source time zone. If the parameter is not specified, it indicates the time zone of the local host.
DestKey	string	Yes	The destination field.
DestFormat	string	Yes	The destination time format in Go.
DestLocation	int	No	The destination time zone. If the parameter is not specified, it indicates the time zone of the local host.
SetTime	bool	No	Specifies whether to overwrite the original time. Default value: true.
KeepSource	bool	No	Specifies whether to return the SourceKey parameter. Default value: true.
NoKeyError	bool	No	Specifies whether to report an error if no match exists. Default value: true.
AlarmIfFail	bool	No	Specifies whether to send alerts when extraction failed. Default value: true.

Use `2006-01-02 15:04:05` (UTC+8) of the `s_key` field as the source time. Convert the time into `2006/01/02 15:04:05` (UTC+9) and save the new time in the `d_key` field. The following example shows how to use this method to process logs.

- Input data

```
"s_key":"2019-07-05 19:28:01"
```

- Configurations

```
{
  "processors":[
    {
      "type":"processor_gotime",
      "detail": {
        "SourceKey": "s_key",
        "SourceFormat":"2006-01-02 15:04:05",
        "SourceLocation":8,
        "DestKey":"d_key",
        "DestFormat":"2006/01/02 15:04:05",
        "DestLocation":9,
        "SetTime": true,
        "KeepSource": true,
        "NoKeyError": true,
        "AlarmIfFail": true
      }
    }
  ]
}
```

- Output data

```
"s_key":"2019-07-05 19:28:01"
"d_key":"2019/07/05 20:28:01"
```

Expand log fields (JSON)

You can use this method to expand a log field.

The type of the plug-in is `processor_json`.

> **Note** This plug-in is available for Logtail 0.16.28 or later.

Parameters

Parameter	Type	Required	Description
SourceKey	string	Yes	The name of the field you want to expand.
NoKeyError	bool	No	Specifies whether to report an error if no match exists. Default value: true.
ExpandDepth	int	No	The depth of JSON expansion. The value must be a non-negative integer. Default value: 0. This indicates the depth is not limited. The value n specifies that the number of levels to expand is n.

Parameter	Type	Required	Description
ExpandConnector	string	No	The delimiter that you use to connect expanded levels. Default value: `_`. You can leave this parameter unspecified.
Prefix	string	No	The prefix that you want to add to the name of each new field after expansion.
KeepSource	bool	No	Specifies whether to return the SourceKey parameter. Default value: true.
UseSourceKeyAsPrefix	bool	No	Specifies whether to retain the name of the source field as the prefix of each new field after expansion. Default value: false.

The following example shows how to use the field expansion (JSON) method to expand the `s_key` field.

- Input data

```
"s_key":"{\"k1\":{\"k2\":{\"k3\":{\"k4\":{\"k51\":\"51\",\"k52\":\"52\"},\"k41\":\"41\"}}})"
```

- Configurations

```
{
  "processors":[
    {
      "type":"processor_json",
      "detail": {
        "SourceKey": "s_key",
        "NoKeyError":true,
        "ExpandDepth":0,
        "ExpandConnector":"-",
        "Prefix":"j",
        "KeepSource": false,
        "UseSourceKeyAsPrefix": true
      }
    }
  ]
}
```

- Output data

```
"s_key":"{\"k1\":{\"k2\":{\"k3\":{\"k4\":{\"k51\":\"51\",\"k52\":\"52\"},\"k41\":\"41\"}}})"
"js_key-k1-k2-k3-k4-k51":"51"
"js_key-k1-k2-k3-k4-k52":"52"
"js_key-k1-k2-k3-k41":"41"
```

Combine log fields (JSON)

You can use this method to combine multiple log fields into one field.

The type of the plug-in is `processor_packjson`.

> **Note** This plug-in is available for Logtail 0.16.28 or later.

Parameters

Parameter	Type	Required	Description
SourceKeys	array	Yes	The array that includes the names of fields that you want to combine.
DestKey	string	No	The name of the destination field after combination.
KeepSource	bool	No	Specifies whether to return the SourceKey parameter. Default value: true.
AlarmIfIncomplete	bool	No	Specifies whether to send alerts if no source fields exist. Default value: true.

The following example shows how to use the method to combine the `a` and `b` fields into the `d_key` field.

- Input data

```
"a":"1"
"b":"2"
```

- Configurations

```
{
  "processors":[
    {
      "type":"processor_packjson",
      "detail": {
        "SourceKeys": ["a","b"],
        "DestKey":"d_key",
        "KeepSource":true,
        "AlarmIfEmpty":true
      }
    }
  ]
}
```

- Output data

```
"a":"1"
"b":"2"
"d_key":"{\"a\":\"1\",\"b\":\"2\"}"
```

Rename fields

You can use this method to rename multiple fields.

The type of the plug-in is `processor_rename`.

> **Note** This plug-in is available for Logtail 0.16.28 or later.

Parameters

Parameter	Type	Required	Description
NoKeyError	bool	No	Specifies whether to report an error if no match exists. Default value: true.

Parameter	Type	Required	Description
SourceKeys	array	Yes	The array that includes the names of fields that you want to rename.
DestKeys	array	Yes	The array that includes the new names of the fields.

The following example shows how to use this method to rename the `aaa1` and `aaa2` fields to `bbb1` and `bbb2` fields.

- Input data

```
"aaa1":"value1"
"aaa2":"value2"
"aaa3":"value3"
```

- Configurations

```
{
  "processors":[
    {
      "type":"processor_rename",
      "detail": {
        "SourceKeys": ["aaa1","aaa2"],
        "DestKeys": ["bbb1","bbb2"],
        "NoKeyError": true
      }
    }
  ]
}
```

- Output data

```
"bbb1":"value1"
"bbb2":"value2"
"aaa3":"value3"
```

Extract log time (Strptime)

You can use this method to extract time from a log field and parse the time by using the Linux strptime() function.

The type of the plug-in is `processor_strptime`.

> **Note** This plug-in is available for Logtail 0.16.28 or later.

Parameters

Parameter	Type	Required	Description
SourceKey	string	Yes	The name of the field from which you want to extract time.
Format	string	Yes	The time format to parse the time.
AdjustUTCOffset	bool	No	Specifies whether to adjust the time zone. Default value: false.

Parameter	Type	Required	Description
UTCOffset	int	No	The offset that is used to adjust the time zone. Unit: seconds. For example, if you set the value to 14400, the time zone is changed to UTC+8.
AlarmIfFail	bool	No	Specifies whether to send alerts if a field fails to be extracted.
KeepSource	bool	No	Specifies whether to return the SourceKey parameter. Default value: true.

Parse the value of the `log_time` in the `%Y/%m/%d %H:%M:%S` time format and use the time zone of your local host. The following examples show how to use this method to process logs.

- Example 1: The time zone is UTC+8.
  - Input data

    ```
    "log_time":"2016/01/02 12:59:59"
    ```

  - Configurations

    ```
    {
      "processors":[
        {
          "type":"processor_strptime",
          "detail": {
            "SourceKey": "log_time",
            "Format": "%Y/%m/%d %H:%M:%S"
          }
        }
      ]
    }
    ```

  - Output data

    ```
    "log_time":"2016/01/02 12:59:59"
    Log.Time = 1451710799
    ```

- Example 2: The time zone is UTC+7.
  - Input data

    ```
    "log_time":"2016/01/02 12:59:59"
    ```

- Configurations

```
{
  "processors":[
    {
      "type":"processor_strptime",
      "detail": {
        "SourceKey": "log_time",
        "Format": "%Y/%m/%d %H:%M:%S",
        "AdjustUTCOffset": true,
        "UTCOffset": 25200
      }
    }
  ]
}
```

- Output data

```
"log_time":"2016/01/02 12:59:59"
Log.Time = 1451714399
```

Custom methods

You can use a combination of multiple processing methods to process logs. The following example shows how to use a single-character delimiter to split a log into several fields and then specify anchor points to extract content from the `detail` field.

- Input data

```
"content" :
"ACCESS|QAS|11. **. **.**|1508729889935|52460dbed4d540b88a973cf5452b1447|1238|appKey=ba,env=pub,requestTime=1508729889913,latency=22ms,
request={appKey:ba,optional:{\\domains\\:\\daily\\,\\version\\:\\v2\\},rawQuery:{\\query\\:\\The route to Location A\\,\\domain\\:\\Navigation\\,\\intent\\:\\navigate\\,\\slots\\:\\to_geo:level3=Location A\\,\\location\\:\\Location B\\},
requestId:52460dbed4d540b88a973cf5452b1447},
response={answers:[],status:SUCCESS}|"
```

- Configurations

```
"processors" : [
      {
          "type" : "processor_split_char",
          "detail" : {"SourceKey" : "content",
              "SplitSep" : "|",
              "SplitKeys" : ["method", "type", "ip", "time", "req_id", "size", "detail"]
          }
      },
      {
          "type" : "processor_anchor",
          "detail" : "SourceKey" : "detail",
              "Anchors" : [
                  {
                          "Start" : "appKey=",
                      "Stop" : ",env=",
                      "FieldName" : "appKey",
                      "FieldType" : "string"
                  },
                  {
                      "Start" : ",env",
                      "Stop" : ",requestTime=",
                      "FieldName" : "env",
                      "FieldType" : "string"
                  },
                  {
                      "Start" : ",requestTime=",
                      "Stop" : ",latency",
                      "FieldName" : "requestTime",
                      "FieldType" : "string"
                  },
                  {
                      "Start" : ",latency=",
                      "Stop" : ",request=",
                      "FieldName" : "latency",
                      "FieldType" : "string"
                  },
                  {
                      "Start" : ",request=",
                      "Stop" : ",response=",
                      "FieldName" : "request",
                      "FieldType" : "string"
                  },
                  {
                      "Start" : ",response=",
                      "Stop" : "",
                      "FieldName" : "response",
                      "FieldType" : "json"
                  }
              ]
          }
      }
  ]
```

- Output data

```
"method" : "ACCESS"
"type" : "QAS"
"ip" : "**. **. **.**"
"time" : "1508729889935"
"req_id" : "52460dbed4d540b88a973cf5452b1447"
"size" : "1238"
"appKey" : "ba"
"env" : "pub"
"requestTime" : "1508729889913"
"latency" : "22ms"
"request" : "{appKey:nui-banma,optional:{\\domains\\:\\daily-faq\\,\\version\\:\\v2\\},rawQuery:{\
\query\\:\\\345\216\273\344\271\220\345\261\261\347\232\204\350\267\257\347\272\277\\,\\domain\\:\
\\345\257\274\350\210\252\\,\\intent\\:\\navigate\\,\\slots\\:\\to_geo:level3=\344\271\220\345\261
\261\\,\\location\\:\\\345\214\227\344\272\254\\},requestId:52460dbed4d540b88a973cf5452b1447}"
"response_answers" : "[]"
"response_status" : "SUCCESS"
```

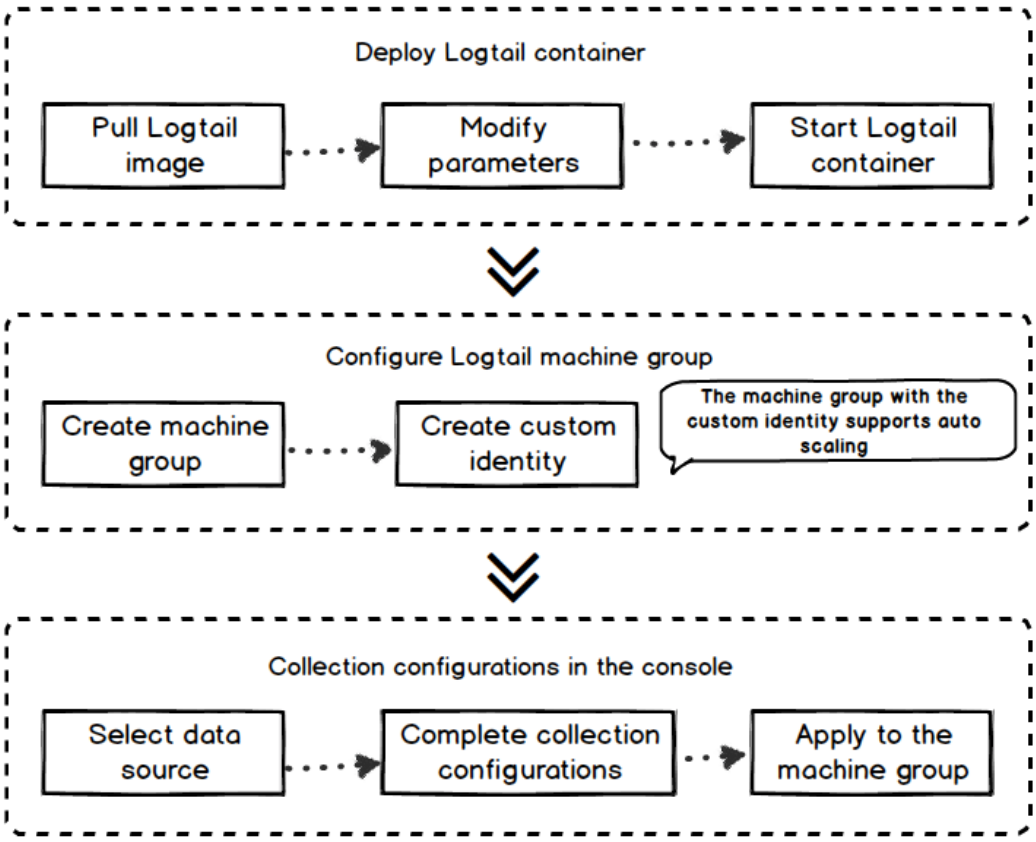

28.1.3.1.6. Collect container logs

28.1.3.1.6.1. Collect standard Docker logs

This topic describes how to use Logtail to collect standard Docker logs and upload these logs together with the container metadata to Log Service.

Procedure

Procedure



1. Deploy a Logtail container.

2. Configure a Logtail server group.

   Create a server group with a custom ID in the Log Service console. The container cluster can automatically scale up or down based on data traffic.

3. Create a Logtail configuration.

   Create a Logtail configuration in the Log Service console. The Logtail configuration process is completed in the Log Service console. No local configuration is needed.

Deploy a Logtail container

1. Run the following command to pull the Logtail image.

```
docker pull registry.cn-hangzhou.aliyuncs.com/log-service/logtail
```

2. Start a Logtail container.

   Set the `${your_region_name}`, `${your_aliyun_user_id}`, and `${your_machine_group_user_defined_id}` parameters in the startup template.

```
docker run -d -v /:/logtail_host:ro -v /var/run:/var/run --env
ALIYUN_LOGTAIL_CONFIG=/etc/ilogtail/conf/${your_region_name}/ilogtail_config.json
--env ALIYUN_LOGTAIL_USER_ID=${your_aliyun_user_id} --env
 ALIYUN_LOGTAIL_USER_DEFINED_ID=${your_machine_group_user_defined_id} registry.cn-hangzhou.aliyun
cs.com/log-service/logtail
```

   Notice Before you set the parameters, you must complete one of the following configurations. Otherwise, the `container text file busy` error may occur when you delete another container.

   - For CentOS 7.4 and later versions, set fs.may_detach_mounts to 1. For more information, see Bug 1468249, Bug 1441737, and Issue 34538.
   - Grant the `privileged` permission to Logtail by adding the `--privileged` flag to the startup parameters. For more information, see Docker run reference.

Parameter	Description
`${your_region_name}`	The region of the project. For more information, see View the information of a project.
`${your_aliyun_user_id}`	The user ID. Set this parameter to the ID of your Alibaba Cloud account, which is a string. For information about how to view the ID, see Step 1 in Configure an account ID for a server.
`${your_machine_group_user_defined_id}`	The custom ID of your server group. For information about how to set the custom ID, see Step 1 in Create a machine group based on a custom ID.

   After you set the parameters, run the following command to start the Logtail container.

```
docker run -d -v /:/logtail_host:ro -v /var/run:/var/run
--env ALIYUN_LOGTAIL_CONFIG=/etc/ilogtail/conf/cn_hangzhou/ilogtail_config.json --env
 ALIYUN_LOGTAIL_USER_ID=1654218******--env ALIYUN_LOGTAIL_USER_DEFINED_ID=log-docker-demo registr
y.cn-hangzhou.aliyuncs.com/log-service/logtail
```

> **Notice**
>
> You can customize the startup parameters of the Logtail container if the following conditions are met:
>
> - The following environment variables exist before you start the Logtail container: `ALIYUN_LOGTAIL_USER_DEFINED_ID`, `ALIYUN_LOGTAIL_USER_ID`, and `ALIYUN_LOGTAIL_CONFIG`.
> - The */var/run* directory is mounted on the */var/run* directory of the Logtail container.
> - To collect container standard output, container logs, or host files, you must mount the root directory on the `/logtail_host` directory of the Logtail container.
> - If an error showing *The parameter is invalid : uuid=none* occurs in the `/usr/local/ilogtail/ilogtail.LOG` Logtail log file, create a file named *product_uuid* on the host. Add a valid UUID such as `169E98C9-ABC0-4A92-B1D2-AA6239C0D261` to the file, and mount the file on the */sys/class/dmi/id/product_uuid* directory of the Logtail container.

Configure a Logtail server group

1. Log on to the Log Service console.
2. Click a project name.
3. In the left-side navigation pane, click the **Server Groups** icon to show the server group list.
4. Click the icon next to Server Groups, and then select **Create Server Group**.

   You can also create a server group when you import data to Log Service.
5. In the dialog box that appears, select **Custom ID** from the Identifier drop-down list. Enter the value of `ALIYUN_LOGTAIL_USER_DEFINED_ID` set in the previous step in the **Custom Identifier** field.

Click OK. One minute later, click the name of the server group in the **Server Groups** list. On the **Server Group Settings** page that appears, you can view the heartbeat status of the Logtail container. For more information, see View the status of a server group.

Create a Logtail configuration

Create a Logtail configuration in the console.

- For more information about Docker logs, see Collect container text logs.
- For more information about Docker standard output, see Collect stdout and stderr logs from containers.
- Host text logs.

  The root directory of a host is mounted on the `/logtail_host` directory of the Logtail container by default. You must add the /logtail_host prefix to the log path. For example, if you want to collect data from the `/home/logs/app_log/` directory of the host, you must set the log path as `/logtail_host/home/logs/app_log/`.

What to do next

- View the status of the Logtail container.

  You can run the `docker exec ${logtail_container_id} /etc/init.d/ilogtaild status` command to view the status of Logtail.
- View the version number, IP address, and startup time of Logtail.

  You can run the `docker exec ${logtail_container_id} cat /usr/local/ilogtail/app_info.json` command to view Logtail information.
- View the operations logs of Logtail.

  The operations logs of Logtail are stored in the `ilogtail.LOG` file in the `/usr/local/ilogtail/` directory. If the log file is rotated and compressed, it is stored as a file named `ilogtail.LOG.x.gz`.

For example:

```
[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 tail -n 5 /usr/local/ilogtail/ilogtail.LOG
[2018-02-06 08:13:35.721864]    [INFO]    [8]    [build/release64/sls/ilogtail/LogtailPlugin.cpp:104]    logtail plugin Resume:start
[2018-02-06 08:13:35.722135]    [INFO]    [8]    [build/release64/sls/ilogtail/LogtailPlugin.cpp:106]    logtail plugin Resume:success
[2018-02-06 08:13:35.722149]    [INFO]    [8]    [build/release64/sls/ilogtail/EventDispatcher.cpp:369]    start add existed check point events, size:0
[2018-02-06 08:13:35.722155]    [INFO]    [8]    [build/release64/sls/ilogtail/EventDispatcher.cpp:511]    add existed check point events, size:0    cache size:0    event size:0    success count:0
[2018-02-06 08:13:39.725417]    [INFO]    [8]    [build/release64/sls/ilogtail/ConfigManager.cpp:3776]    check container path update flag:0    size:1
```

Ignore the following standard output:

```
start umount useless mount points, /shm$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fbe2bdb95d13b1e110172ef57fe840c82155/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44beab6e69718/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640b1e16c22dbe/merged: must be superuser to unmount
...
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
ilogtail is running
```

- Restart Logtail.

  To restart Logtail, use the following sample code:

```
[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 /etc/init.d/ilogtaild stop
kill process Name: ilogtail pid: 7
kill process Name: ilogtail pid: 8
stop success
[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 /etc/init.d/ilogtaild start
ilogtail is running
```

28.1.3.1.6.2. Collect Kubernetes logs

This topic describes how to install and use Logtail to collect logs from Kubernetes clusters.

Configuration procedure

Perform the following steps to collect logs from Kubernetes clusters:

1. Install the alibaba-log-controller Helm package.
2. Use the Log Service console to manage log collection configurations.

Step 1: Install Logtail

- Install Logtail in an Alibaba Cloud Container Service for Kubernetes cluster.

  If Log Service components are not installed in your cluster, you must manually install the components.

  i. Connect to the Kubernetes cluster by using CloudShell.

ii. Run the following command in CloudShell to obtain the ID of your Apsara Stack tenant account.

```
echo $ALIBABA_CLOUD_ACCOUNT_ID
```

iii. After you set the `${your_k8s_cluster_id}`, `${your_ali_uid}`, and `${your_k8s_cluster_region_id}` parameters, run the following command:

```
wget https://acs-logging.oss-cn-hangzhou.aliyuncs.com/alicloud-k8s-log-installer.sh -O alicloud-k8s-log-installer.sh; chmod 744 ./alicloud-k8s-log-installer.sh; ./alicloud-k8s-log-installer.sh --cluster-id ${your_k8s_cluster_id} --ali-uid ${your_ali_uid} --region-id ${your_k8s_cluster_region_id}
```

- Install Logtail in a user-created Kubernetes cluster.

> **Notice**
> - The version of the Kubernetes cluster must be 1.8 or later.
> - Helm 2.6.4 or later must be installed.

i. In the Log Service console, create a project whose name starts with `k8s-log-custom-`.

ii. Replace the parameters in the following command based on your business requirements:

```
wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/kubernetes/alicloud-log-k8s-custom-install.sh; chmod 744 ./alicloud-log-k8s-custom-install.sh; sh ./alicloud-log-k8s-custom-install.sh {your-project-suffix} {region-id} {aliuid} {access-key-id} {access-key-secret}
```

The following table lists the parameters in the preceding command.

Parameter	Description
{your-project-suffix}	The portion of the project name at the end of `k8s-log-custom-`. For example, if you create a project whose name is `k8s-log-custom-xxxx`, set this parameter to `xxxx`.
{regionId}	The ID of the region where the project resides. For more information, see View the information of a project.
{aliuid}	The user ID. Set this parameter to the ID of your Apsara Stack tenant account. **Note** The ID of an Apsara Stack tenant account is a string. For more information about how to obtain the ID, see Configure an account ID for a server.
{access-key-id}	The AccessKey ID of your Apsara Stack tenant account.
{access-key-secret}	The AccessKey secret of your Apsara Stack tenant account.

After Logtail is installed in the Kubernetes cluster, Log Service automatically creates a machine group named `k8s-group-${your_k8s_cluster_id}` for the project.

> **Note**
>
> - A Logstore named `config-operation-log` is automatically created in the project. Do not delete the Logstore.
> - When you install Logtail in a user-created Kubernetes cluster, Logtail is granted `privileged` permissions by default. This prevents the `container text file busy` error when you delete a pod. For more information, visit Bug 1468249, Bug 1441737, and Issue 34538.

The following example shows a successful installation:

```
[root@iZbp1dsxxxxxqfbiaZ ~]#  wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com
/kubernetes/alicloud-log-k8s-custom-install.sh; chmod 744 ./alicloud-log-k8s-custom-install.sh; sh
./alicloud-log-k8s-custom-install.sh xxxx cn-hangzhou 165xxxxxxxx050 LTAxxxxxxxxxxx AIxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxe
....
....
....
NAME:   alibaba-log-controller
LAST DEPLOYED: Fri May 18 16:52:38 2018
NAMESPACE: default
STATUS: DEPLOYED
RESOURCES:
==> v1beta1/ClusterRoleBinding
NAME                    AGE
alibaba-log-controller  0s
==> v1beta1/DaemonSet
NAME        DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE  NODE SELECTOR  AGE
logtail-ds  2        2        0      2           0          <none>         0s
==> v1beta1/Deployment
NAME                    DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
alibaba-log-controller  1        1        1           0          0s
==> v1/Pod(related)
NAME                                    READY  STATUS             RESTARTS  AGE
logtail-ds-7xf2d                        0/1    ContainerCreating  0         0s
logtail-ds-9j4bx                        0/1    ContainerCreating  0         0s
alibaba-log-controller-796f8496b6-6jxb2  0/1    ContainerCreating  0         0s
==> v1/ServiceAccount
NAME                    SECRETS  AGE
alibaba-log-controller  1        0s
==> v1beta1/CustomResourceDefinition
NAME                                  AGE
aliyunlogconfigs.log.alibabacloud.com  0s
==> v1beta1/ClusterRole
alibaba-log-controller  0s
[INFO] your k8s is using project : k8s-log-custom-xxx, region : cn-hangzhou, aliuid : ************
***, accessKeyId : LTA**********
[SUCCESS] install helm package : alibaba-log-controller success.
```

To check the status of each Log Service component in the Kubernetes cluster, run the `helm status alibaba-log-controller` command. If all pods are in the Running state, Logtail is installed.

Log on to the Log Service console to find the project. If you have multiple projects, search for the project by using the `k8s-log` keyword.

Step 2: Configure log collection

Create Logtail configurations for log collection in the console as required.

- For information about how to collect Kubernetes text logs, see Collect container text logs.
- For information about how to collect Kubernetes stdout logs, see Collect stdout and stderr logs from containers.
- Host text logs.

  By default, the root directory of a host is mounted to the `/logtail_host` directory of the Logtail container. You must add the /logtail_host prefix to the log path. For example, if you want to collect data from the `/home/logs/app_log/` directory of the host, you must set the log path to `/logtail_host/home/logs/app_log/`.

Other common commands

- Store logs of multiple clusters in one project

  You can collect logs from multiple Kubernetes clusters. If you want to store these logs in the same project, you can specify the same cluster ID for the `${your_k8s_cluster_id}` parameter when you install Log Service components on multiple Kubernetes clusters.

  For example, if you have three Kubernetes clusters whose IDs are abc001, abc002, and abc003, specify `abc001` for the `${your_k8s_cluster_id}` parameter when you install Log Service components for each Kubernetes cluster.

  > **Notice** This feature is unavailable for Kubernetes clusters that reside in different regions.

- Logtail container logs

  Logtail log files named `ilogtail.LOG` and `logtail_plugin.LOG` are stored in the `/usr/local/ilogtail/` directory of a Logtail container. Ignore the following standard output:

  ```
  start umount useless mount points, /shm$|/merged$|/mqueue$
  umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fbe2bdb95d13b1e110172ef5
  7fe840c82155/merged: must be superuser to unmount
  umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff4
  4beab6e69718/merged: must be superuser to unmount
  umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a64
  0b1e16c22dbe/merged: must be superuser to unmount
  ......
  xargs: umount: exited with status 255; aborting
  umount done
  start logtail
  ilogtail is running
  logtail status:
  ilogtail is running
  ```

- View the status of each Log Service component in a Kubernetes cluster

  ```
  helm status alibaba-log-controller
  ```

- Troubleshoot alibaba-log-controller startup failures

  Make sure that the following conditions are met:

  - Log Service components are installed on the master node of the Kubernetes cluster.
  - The Kubernetes cluster ID that you specified is valid when you install Log Service components.

  If Log Service components fail to be installled because the preceding conditions are not met, run the `helm del --purge alibaba-log-controller` command to delete the installation package and install Log Service components again.

- View the status of Logtail DaemonSet in a Kubernetes cluster

  Run the `kubectl get ds -n kube-system` command.

> **Note** The default namespace of Logtail is `kube-system`.

- View the version number, IP address, and startup time of Logtail.

  Example:

```
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl get po -n kube-system | grep logtail
NAME            READY     STATUS    RESTARTS   AGE
logtail-ds-gb92k   1/1       Running   0          2h
logtail-ds-wm7lw   1/1       Running   0          4d
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-ds-gb92k -n kube-system cat /usr/local/ilog
tail/app_info.json
{
   "UUID" : "",
   "hostname" : "logtail-ds-gb92k",
   "instance_id" : "0EBB2B0E-0A3B-11E8-B0CE-0A58AC140402_172.20.4.2_1517810940",
   "ip" : "172.20.4.2",
   "logtail_version" : "0.16.2",
   "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
   "update_time" : "2018-02-05 06:09:01"
}
```

- View the operational logs of Logtail

  Logtail operational logs are stored in the `ilogtail.LOG` file in the `/usr/local/ilogtail/` directory. If the log file is rotated and compressed, it is stored as a file named `ilogtail.LOG.x.gz`.

  Example:

```
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-ds-gb92k -n kube-system tail /usr/local/ilo
gtail/ilogtail.LOG
[2018-02-05 06:09:02.168693] [INFO] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:104] logta
il plugin Resume:start
[2018-02-05 06:09:02.168807] [INFO] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:106] logta
il plugin Resume:success
[2018-02-05 06:09:02.168822] [INFO] [9] [build/release64/sls/ilogtail/EventDispatcher.cpp:369] sta
rt add existed check point events, size:0
[2018-02-05 06:09:02.168827] [INFO] [9] [build/release64/sls/ilogtail/EventDispatcher.cpp:511] add
existed check point events, size:0 cache size:0 event size:0 success count:0
```

- Restart Logtail in a pod

  Example:

```
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-ds-gb92k -n kube-system /etc/init.d/ilogtai
ld stop
kill process Name: ilogtail pid: 7
kill process Name: ilogtail pid: 9
stop success
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-ds-gb92k -n kube-system /etc/init.d/ilogtai
ld start
ilogtail is running
```

28.1.3.1.6.3. Collect container text logs

Logtail collects text logs generated in containers and uploads these logs together with the container-related metadata information to Log Service.

Features

Compared with basic log file collection, Docker file collection has the following characteristics:

- Allows you to configure the log path of a container, without the need to consider the mapping between the path and the host.
- Allows you to use labels to specify the containers whose logs are to be collected.
- Allows you to use labels to exclude specific containers.
- Allows you to use environment variables to specify the containers whose logs are to be collected.
- Allows you to use environment variables to exclude specific containers.
- Supports multiline logs such as Java Stack logs.
- Supports automatic labeling for container data.
- Supports automatic labeling for Kubernetes containers.

Limits

- Collection stop policy: When a container is stopped and Logtail detects the `die` event on the container, Logtail stops collecting logs of the container (with a latency of no more than 3 seconds). In this case, if a collection latency occurs, some logs generated before the stop action may be lost.
- Docker storage driver: Only overlay and overlay2 are supported. For other storage drivers, you must mount the log directory to the local host.
- Logtail running mode: Logtail must run in a container and must be deployed based on Logtail deployment solutions.
- Label: refers to the label information in docker inspect. It is not synonymous with labels in Kubernetes.
- Environment: refers to the environment information configured during container startup.

Procedure

1. Deploy and configure the Logtail container.
2. Configure log collection in Log Service.

Logtail deployment and configuration

- Kubernetes

  For more information about Kubernetes log collection, see Logtail deployment solution for collecting Kubernetes logs.

- Management methods for other containers

  For more information about management methods for other containers, such as Swarm and Mesos, see Common deployment solution for collecting Docker logs.

Collection configuration

1. Log on to the Log Service console.
2. Click the **Import Data** button. On the **Import Data** page that appears, select **Docker File**.
3. Select a destination project and Logstore, and then click **Next**.

   You can also click **Create Now** to create a project and a Logstore.

   If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

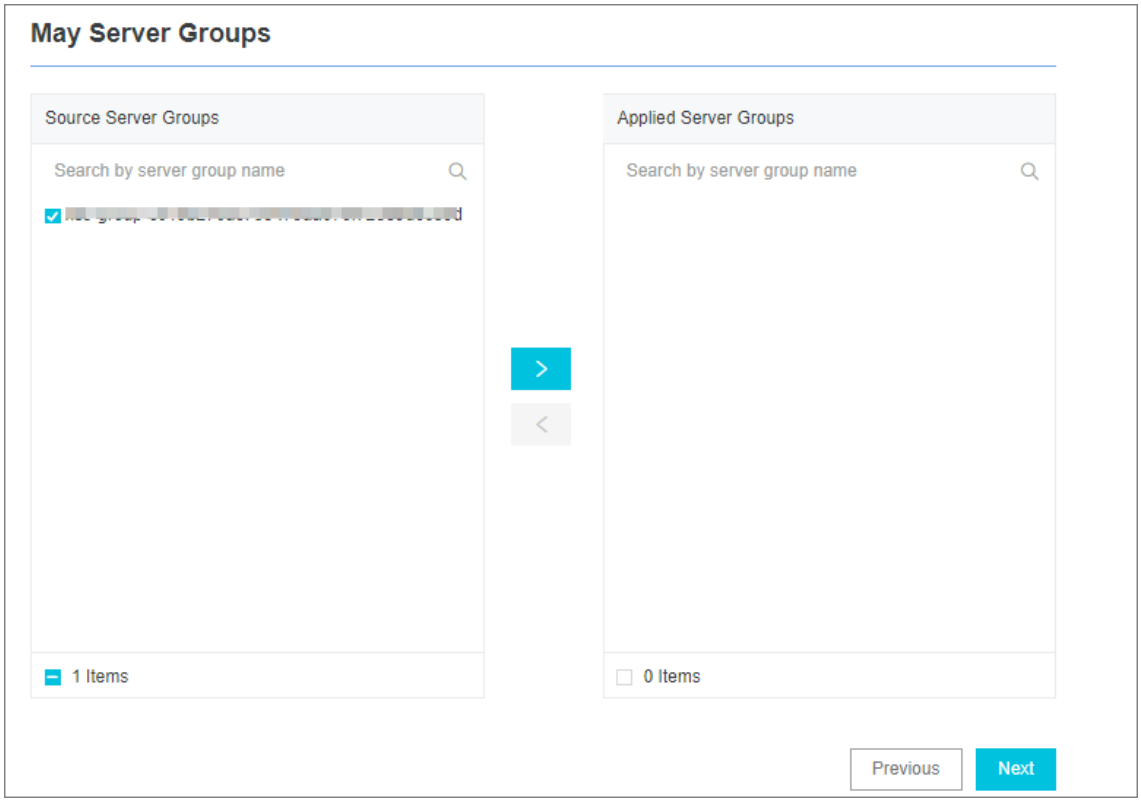
4. Create a machine group and click **Next**.

   Before you can create a machine group, you must install Logtail.

   Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

   After you install Logtail, click **Complete Installation** to create a machine group. If a machine group is created, click **Use Existing Machine Groups** to select the machine group.

5. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



> **Notice** If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

6. Configure Logtail.

   The following table lists data source-specific parameters. For more information about common parameters, see Configure text log collection.

Parameter	Description
Docker File	This parameter is used to check whether the collected target file is a Docker file.
Label Whitelist	LabelKey is required. If LabelValue is not empty, only containers whose labels contain LabelKey = LabelValue are collected. If LabelValue is empty, all the containers whose labels contain LabelKey are collected. **Note** ○ Key-value pairs are disjunctive with each other. If the label of a container contains one of the key-value pairs you specify, logs of the container are collected. ○ Labels refer to Docker labels.

Parameter	Description
Label Blacklist	LabelKey is required. If LabelValue is not empty, only containers whose labels contain LabelKey = LabelValue are excluded. If LabelValue is empty, all containers whose labels contain LabelKey are excluded. **Note** ○ Key-value pairs are disjunctive with each other. If the label of a container contains one of the key-value pairs you specify, the container is excluded. ○ Labels described in this topic refer to the label information in docker inspect.
Environment Variable Whitelist	EnvKey is required. If EnvValue is not empty, only containers whose environment variables contain EnvKey = EnvValue are collected. If EnvValue is empty, all containers whose environment variables contain EnvKey are collected. **Note** ○ Key-value pairs are disjunctive with each other. If the environment variable of a container contains one of the key-value pairs you specify, logs of the container are collected. ○ The environment variable refers to the environment information configured in container startup.
Environment Variable Blacklist	EnvKey is required. If EnvValue is not empty, only containers whose environment variables contain EnvKey = EnvValue are excluded. If EnvValue is empty, all containers whose environment variables contain EnvKey are excluded. **Note** ○ Key-value pairs are disjunctive with each other. If the environment variable of a container contains one of the key-value pairs you specify, the container is excluded. ○ The environment variable refers to the environment information configured in container startup.

Note

- Labels in whitelist and blacklist are different from those defined in Kubernetes. Labels in this topic refer to the label information in docker inspect.
- A namespace and a container name in Kubernetes can be mapped to Docker labels. LabelKey corresponding to a namespace is `io.kubernetes.pod.namespace`. LabelKey corresponding to a container name is `io.kubernetes.container.name`. For example, the namespace of the pod you created is backend-prod and the container name is worker-server. In this case, you can configure a whitelist label: `io.kubernetes.pod.namespace : backend-prod` or `io.kubernetes.container.name : worker-server`, so that only logs of the container are collected.
- In Kubernetes, we recommend that you only use the `io.kubernetes.pod.namespace` and `io.kubernetes.container.name` labels. In other cases, use an environment variable whitelist or blacklist.

7. Configure indexes in the Configure Query and Analysis step. Click **Next**.

By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Enable the index feature and configure indexes for a Logstore.

> **Note**
> - To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings of Field Search prevail.
> - If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

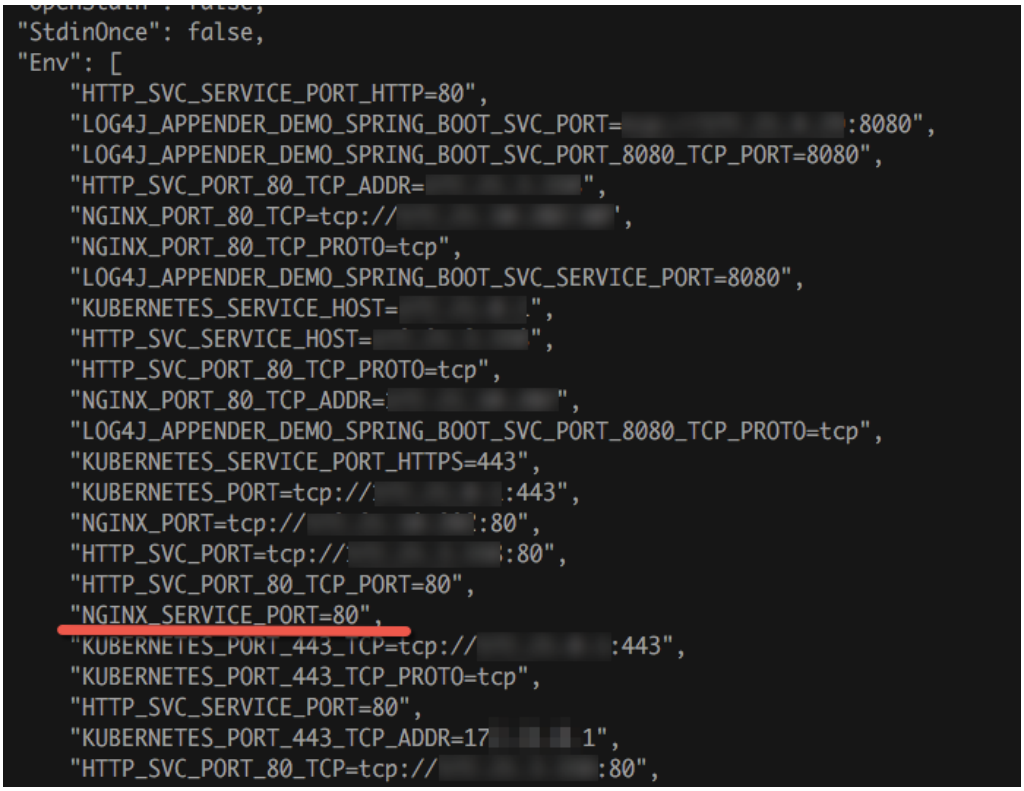
Configuration examples

- Environment configuration

  Collect the logs of the container whose environment variable is `NGINX_PORT_80_TCP_PORT=80` but not `POD_NAMESPACE=kube-system`. The log file path is `/var/log/nginx/access.log` and logs are parsed in simple mode.

  > **Note** The environment variable refers to the environment information configured in container startup.

  Environment configuration example



- Label configuration

  Collect the logs of a container that meets the following conditions: the container label is `io.kubernetes.container.name=nginx`, the log file path is `/var/log/nginx/access.log`, and logs are parsed in simple mode.

  > **Note** Labels refer to Docker labels.

  Label configuration example

Default fields

Each uploaded log of a common Docker container contains the following fields.

Field	Description:
_image_name_	The name of the image.
_container_name_	The name of the container.
_container_ip_	The IP address of the container.

If a Kubernetes cluster is used, each uploaded log contains the following fields.

Field	Description
_image_name_	The name of the image.
_container_name_	The name of the container.
_pod_name_	The name of the pod.
namespace	The namespace to which the pod belongs.
_pod_uid_	The unique identifier of the pod.
_container_ip_	The IP address of the pod.

28.1.3.1.6.4. Collect stdout and stderr logs from containers

This topic describes how to use Logtail to collect standard output (stdout) and standard error (stderr) logs from containers. After you collect stdout and stderr logs, you can upload the logs together with the container-related metadata to Log Service.

Features

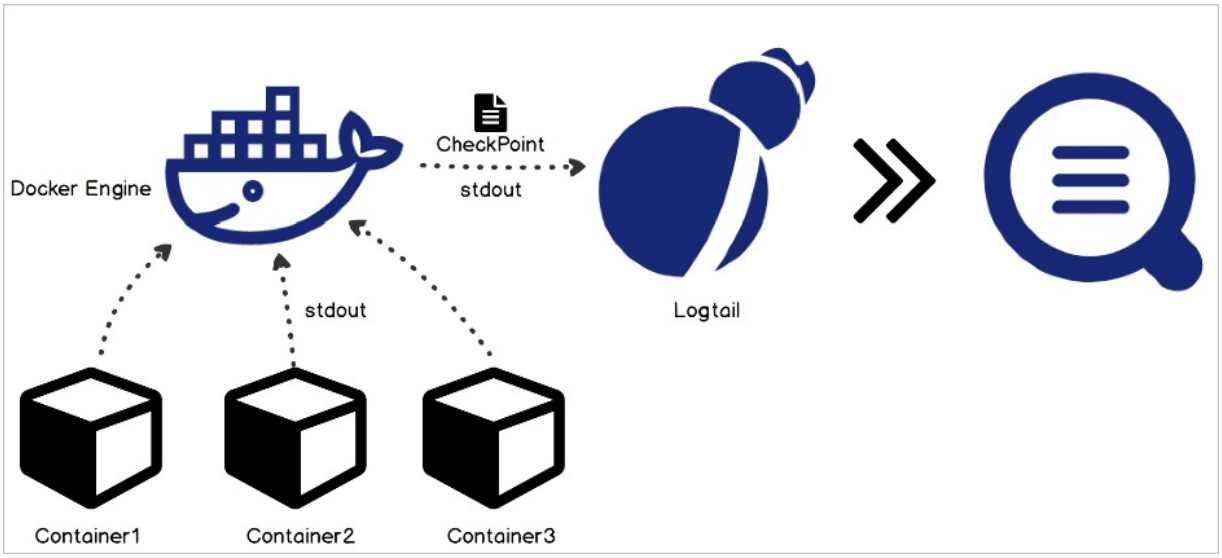
- Collects stdout and xxx logs.
- Labels the containers from which you want to collect stdout and stderr logs.
- Uses tags to exclude specific containers.
- Allows you to use environment variables to specify the containers whose logs are to be collected.
- Allows you to use environment variables to exclude specific containers.
- Supports multiline logs such as Java Stack logs.

- Supports automatic labeling for container data.
- Supports automatic labeling for Kubernetes containers.

Implementation

As shown in the following figure, Logtail communicates with the domain socket of the Docker engine to query containers that run on the Docker engine. Logtail also locates containers from which you want to collect logs based on the specified labels and environment variables. Logtail then uses the docker logs command to collect logs from the specified containers.

When Logtail collects the stdout logs of a container, Logtail records information about log file positions to the checkpoint file at regular intervals. If Logtail is restarted, Logtail collects logs from the last log file position.



Limits

- This feature is available only for Logtail 0.16.0 or later that runs on Linux. For more information about Logtail versions and version updates, see Install Logtail in Linux.
- The domain socket must exist and can access the Docker engine. Otherwise, Logtail cannot access the Docker engine by running the `/var/run/docker.sock` file.
- The last multiline log that you collect must be cached for at least 1,000 seconds. By default, the retention period for a multiline log is 3 seconds. You can specify the period by configuring the `BeginLineTimeoutMs` parameter. The value of the parameter must be a minimum of 1,000 ms. Otherwise, a false positive error may occur.
- Collection stop policy: When a container is stopped and Logtail detects the `die` event on the container, Logtail stops collecting stdout logs of the container. In this case, if a collection latency occurs, some stdout logs that are generated before the stop action may be lost.
- Docker log driver: Only log drivers of the JSON type are supported in the collection of stdout logs.
- Context: By default, a collection configuration is in the same context. If you need to configure different types of containers in different contexts, configure each type separately.
- Data processing: By default, collected logs start with the `content` field. You can apply standard processing methods to these logs. For more information about how to configure one or more processing methods, see Configure data processing methods.
- Label: refers to the label information in docker inspect. It is not related to labels in Kubernetes configurations.
- Environment: refers to environment information that you specified during container startup.

Procedure

1. Deploy and configure Logtail on one or more containers.

2. Create a Logtail configuration and deliver it to Logtail.

Logtail deployment and configuration

- Kubernetes

  For more information about how to collect Kubernetes logs, see Logtail deployment solution for collecting Kubernetes logs.

- Configure Logtail on other containers

  For more information about how to configure Logtail on other containers, such as Swarm and Mesos, see Common Logtail deployment solution for collecting Docker logs.

Configure a data source

1. Log on to the Log Service console.
2. Click **Import Data**. On the **Import Data** page that appears, select **Docker Standard Output**.
3. Select a destination project and Logstore, and then click **Next**.

   You can also click **Create Now** to create a project and a Logstore.

   If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

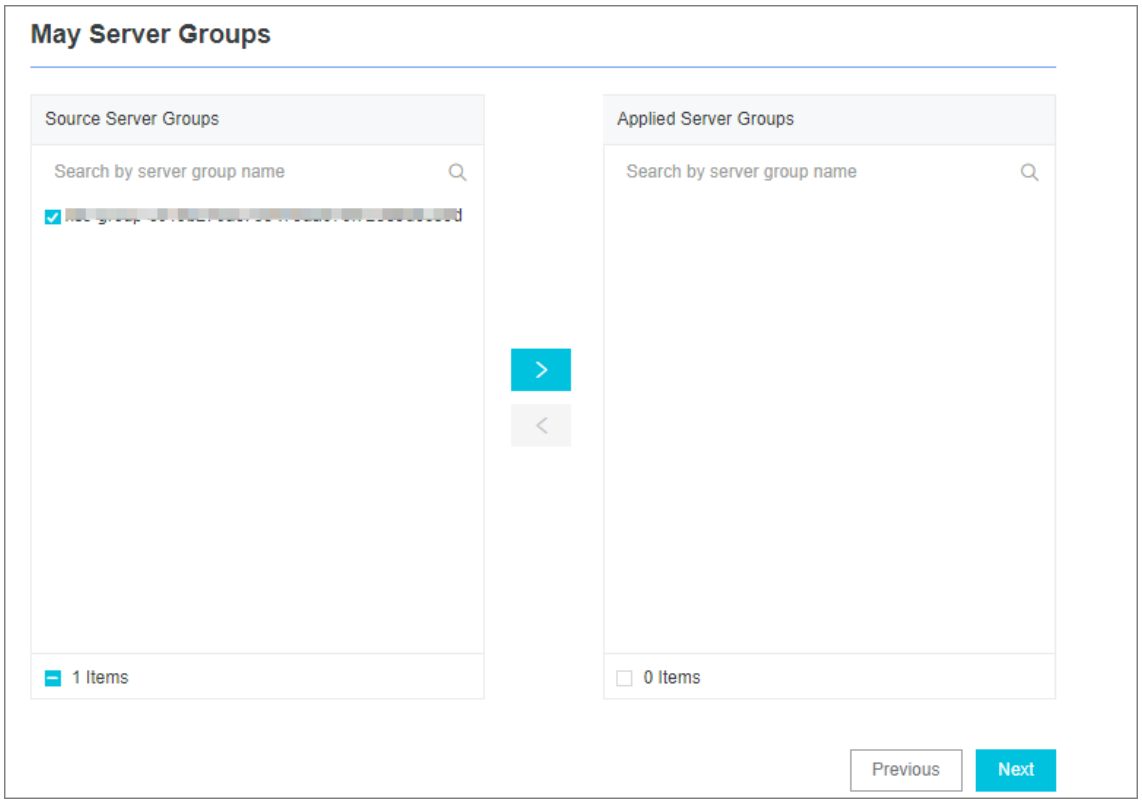
4. Create a machine group and click **Next**.

   Before you can create a machine group, you must install Logtail.

   Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

   After you install Logtail, click **Complete Installation** to create a machine group. If a machine group is created, click **Use Existing Machine Groups** to select the machine group.

5. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



> **Notice** If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

6. Configure a data source.

   In the **Plug-in Config** section, set the required parameters. The following example shows how to set these parameters. For more information, see Parameters.

```
{
 "inputs": [
     {
         "type": "service_docker_stdout",
         "detail": {
             "Stdout": true,
             "Stderr": true,
             "IncludeLabel": {
                 "io.kubernetes.container.name": "nginx"
             },
             "ExcludeLabel": {
                 "io.kubernetes.container.name": "nginx-ingress-controller"
             },
             "IncludeEnv": {
                 "NGINX_SERVICE_PORT": "80"
             },
             "ExcludeEnv": {
                 "POD_NAMESPACE": "kube-system"
             }
         }
     }
 ]
}
```

7. Configure indexes in the Configure Query and Analysis step. Click **Next**.

   By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Enable the index feature and configure indexes for a Logstore.

> **Note**
> - To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings of Field Search prevail.
> - If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

Parameters

The type of the input source is `service_docker_stdout`.

> **Note** Before Logtail uploads data to Log Service, Logtail can process collected data. For more information about processing methods, see Configure data processing methods.

Parameter	Type	Required	Description

Parameter	Type	Required	Description
IncludeLabel	JSON text. Key: JSON string. Value: JSON string.	Yes	By default, this parameter is not specified. If the parameter is not specified, Logtail collects logs from all containers. If you set the key field and leave the value field empty, Logtail collects logs from containers whose labels include this key. **Note** • Key-value pairs are disjunctive with each other. If the label of a container matches one of the key-value pairs, logs of the container are collected. • Labels refer to docker labels.
ExcludeLabel	JSON text. Key: JSON string. Value: JSON string.	No	This parameter is not specified by default. If the parameter is empty, no containers are excluded. If the key is not empty but the value is empty, the containers whose labels include this key are excluded. **Note** • Key-value pairs are disjunctive with each other. If the label of a container matches one of the key-value pairs, the container is excluded. • Labels described in this topic refer to Docker labels.
IncludeEnv	Map. Key: string. Value: string	No	This parameter is empty by default. If the parameter is empty, logs of all containers are collected. If the key is not empty but the value is empty, logs of the containers whose environment variables include this key are collected. **Note** • Key-value pairs are disjunctive with each other. If the environment variable of a container includes one of the key-value pairs, the container is excluded. • The environment variable refers to the environment information configured in container startup.

Parameter	Type	Required	Description
ExcludeEnv	Map. Key: string. Value: string	No	This parameter is empty by default. If you leave the parameter empty, no container is excluded. If you set the key and leave the value empty, the containers whose environment variables include this key are excluded. **Note** • Key-value pairs are disjunctive with each other. When the environment variable of a container includes one of the key-value pairs, the container is excluded. • The environment variable refers to the environment information configured in container startup.
Stdout	Boolean	No	If the value of the parameter is false, stdout data is not collected. Default value: true.
Stderr	Boolean	No	Default value: true. If the value of the parameter is false, stderr data is not collected.
BeginLineRegex	String	No	This parameter is not specified by default. If the parameter is not empty, the regular expression is used to match the first line of each log. If a line matches this regular expression, this line is assumed as the start of a new log. Otherwise, this line is assumed as part of the previous log.
BeginLineTimeoutMs	Integer	No	The timeout period for the regular expression to match a line. Default value: 3000. Unit: ms. If no new log appears within 3 seconds, the most recent log is uploaded.
BeginLineCheckLength	Integer	No	The length of data for the regular expression to match. Default value: 10×1024. Unit: bytes. You can set this parameter to check whether the beginning part of a line can match the regular expression. This improves matching efficiency.
MaxLogSize	Integer	No	The maximum length of a log. Default value: 512×1024. Unit: bytes. If the length exceeds this value, the log data is uploaded directly without finding the first line of logs.

Note

- Labels defined in IncludeLabel and ExcludeLabel are different from those defined in Kubernetes. Labels in this topic refer to Docker labels.
- A namespace and a container name in Kubernetes can be mapped to Docker labels. The LabelKey parameter corresponding to a namespace is `io.kubernetes.pod.namespace`. The LabelKey parameter corresponding to a container name is `io.kubernetes.container.name`. For example, the namespace of the pod you created is backend-prod and the container name is worker-server. In this case, you can configure a whitelist label: `io.kubernetes.pod.namespace : backend-prod` or `io.kubernetes.container.name : worker-server`, so that only logs of the container are collected.
- In Kubernetes, we recommend that you use the `io.kubernetes.pod.namespace` and `io.kubernetes.container.name` labels. In other cases, use IncludeEnv or ExcludeEnv.

Default fields

- Common Docker containers

  Each uploaded log contains the following fields.

Field	Description:
`_time_`	The data upload time. Example: `2018-02-02T02:18:41.979147844Z`.
`_source_`	The type of the input source. Valid values: stdout and stderr.
`_image_name_`	The name of the image.
`_container_name_`	The name of the container.
`_container_ip_`	The IP address of the container.

- Kubernetes containers

  Each uploaded log contains the following fields.

Field	Description
`_time_`	The data upload time. Example: `2018-02-02T02:18:41.979147844Z`.
`_source_`	The type of input sources. Valid values: stdout and stderr.
`_image_name_`	The name of the image.
`_container_name_`	The name of the container.
`_pod_name_`	The name of the pod.
`_namespace_`	The namespace where the pod is located.
`_pod_uid_`	The unique identifier of the pod.
`_container_id_`	The IP address of the pod.

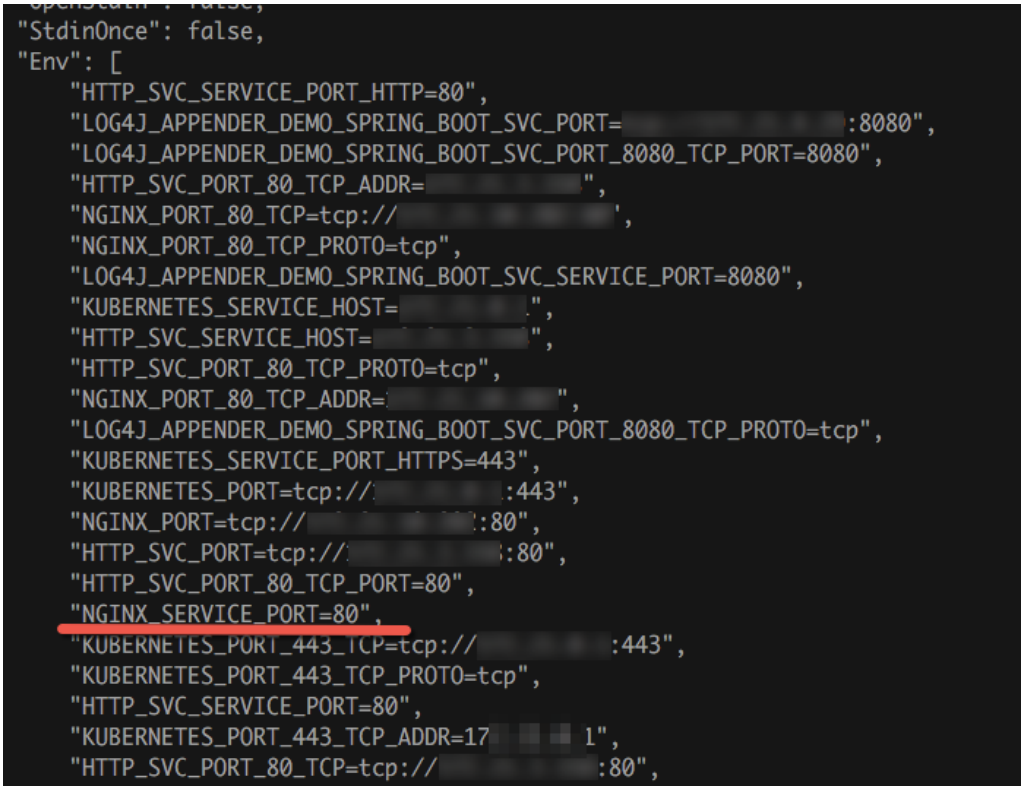
Common configuration examples

- Environment configuration

  Collect the logs of the container whose environment variable is `NGINX_PORT_80_TCP_PORT=80` but not `POD_NAMESPACE=kube-system`.

  > **Note** The environment variable refers to the environment information configured in container startup.

  Environment configuration example



Collection configuration

```
{
    "inputs": [
        {
            "type": "service_docker_stdout",
            "detail": {
                "Stdout": true,
                "Stderr": true,
                "IncludeEnv": {
                    "NGINX_PORT_80_TCP_PORT": "80"
                },
                "ExcludeEnv": {
                    "POD_NAMESPACE": "kube-system"
                }
            }
        }
    ]
}
```

- Label configuration

  Collect the stdout and stderr logs of the container whose label is `io.kubernetes.container.name=nginx` but not `type=pre`.

  > **Note** Labels refer to Docker labels.

  Label configuration example


```
{
    "inputs": [
        {
            "type": "service_docker_stdout",
            "detail": {
                "Stdout": true,
                "Stderr": true,
                "IncludeLabel": {
                    "io.kubernetes.container.name": "nginx"
                },
                "ExcludeLabel": {
                    "type": "pre"
                }
            }
        }
    ]
}
```

Example of configuring multiline log collection

Configuring multiline log collection is important for the collection of Java exception stack logs. The following section introduces a standard collection configuration for Java stdout logs.

- Sample log

```
2018-02-03 14:18:41.968  INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoCo
ntroller : service start
2018-02-03 14:18:41.969 ERROR [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoCo
ntroller : java.lang.NullPointerException
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:19
3)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:199)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96)
...
2018-02-03 14:18:41.968  INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoCo
ntroller : service start done
```

- Collection configuration

  Collect logs of the container whose label is `app=monitor`. The first line of each log to be collected is of the date type. To improve matching efficiency, only the first 10 bytes of each line are checked.

```
{
"inputs": [
  {
    "detail": {
      "BeginLineCheckLength": 10,
      "BeginLineRegex": "\\d+-\\d+-\\d+. *",
      "IncludeLabel": {
        "app": "monitor"
      }
    },
    "type": "service_docker_stdout"
  }
]
}
```

Process collected data

Logtail can process the collected Docker stdout logs by using a common data processing method. Use a regular expression to extract the time, module, thread, class, and info fields.

- Collection configuration

  Collect logs from a container whose label is `app=monitor`. The first line of each log to be collected is of the date type. To improve matching efficiency, only the first 10 bytes of each line are checked.

```
{
"inputs": [
  {
    "detail": {
      "BeginLineCheckLength": 10,
      "BeginLineRegex": "\\d+-\\d+-\\d+. *",
      "IncludeLabel": {
        "app": "monitor"
      }
    },
    "type": "service_docker_stdout"
  }
],
"processors": [
    {
        "type": "processor_regex",
        "detail": {
            "SourceKey": "content",
            "Regex": "(\\d+-\\d+-\\d+ \\d+:\\d+:\\d+\\.\\d+)\\s+(\\w+)\\s+\\[([^]]+)]\\s+\\[([^]]+
)]\\s+:\\s+([\\s\\S]*)",
            "Keys": [
                "time",
                "module",
                "thread",
                "class",
                "info"
            ],
            "NoKeyError": true,
            "NoMatchError": true,
            "KeepSource": false
        }
    }
]
}
```

- Sample output

  After the `2018-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : service start done` log is processed, the following output is returned:

```
__tag__:__hostname__:logtail-dfgef
_container_name_:monitor
_image_name_:registry.cn-hangzhou.aliyuncs.xxxxxxxxxxxxxx
_namespace_:default
_pod_name_:monitor-6f54bd5d74-rtzc7
_pod_uid_:7f012b72-04c7-11e8-84aa-00163f00c369
_source_:stdout
_time_:2018-02-02T14:18:41.979147844Z
time:2018-02-02 02:18:41.968
level:INFO
module:spring-cloud-monitor
thread:nio-8080-exec-4
class:c.g.s.web.controller.DemoController
message:service start done
```

28.1.3.1.7. Limits

This topic describes the limits of Logtail. These limits apply when you collect files, manage resources, and resolve errors.

Limits on file collection

Item	Description
File encoding	Log files encoded in UTF-8 and GBK are supported. We recommend that you use UTF-8 encoding for better processing performance. If log files are encoded in other formats, errors such as garbled characters and data loss may occur.
Log file size	Unlimited.
Log file rotation	Supported. Both `.log*` and `.log` are supported for file names.
Log collection behavior when log parsing is restricted	When log parsing is restricted, Logtail keeps the log file descriptor (FD) open. If log file rotation occurs multiple times during the restriction, Logtail attempts to keep the log parsing sequence of each rotation. If the number of rotated logs to be parsed exceeds 20, Logtail does not process subsequent log files.
Symbolic link	Monitored directories can be soft links.
Size of a single log entry	The maximum size of a single log entry is 512 KB. If a multi-line log entry is divided by using a regular expression to match the first line, the maximum size of each log entry after division is still 512 KB. If the size of a log entry exceeds 512 KB, the log entry is forcibly separated into multiple parts for collection. For example, if the size of a log entry is 1,025 KB, it will be split into three parts: 512 KB, 512 KB, and 1 KB. These log parts are collected in sequence.
Regular expression	Perl-based regular expressions can be used.
Multiple Logtail configuration files for the same log file	Not supported. We recommend that you collect and store log files to one Logstore, and configure multiple subscriptions. If this feature is required, configure symbolic links for log files to bypass this limit.

Item	Description
File opening behavior	When Logtail collects a log file, Logtail opens the log file. If the log file is not modified for more than 5 minutes and log rotation does not occur, Logtail closes the log file.
First log collection behavior	Logtail collects only incremental log files. If a log file is modified for the first time and the log file size exceeds 1 MB, Logtail collects the logs from the last 1 MB. Otherwise, Logtail collects the logs from the beginning of the log file. If the log file is not modified after the Logtail configuration file is sent to the server where the log file resides, Logtail does not collect the log file.
Non-standard text logs	If a log entry contains \0 in multiple lines, the log entry is truncated at the first \0.

Limits on checkpoints

Item	Description
Checkpoint timeout period	If a log file is not modified for more than 30 days, the checkpoint of the log file is deleted.
Checkpoint storage policy	Checkpoints are saved every 15 minutes and are automatically saved when you exit Logtail.
Checkpoint storage path	By default, checkpoints are stored in the `/tmp/logtail_checkpoint` directory. For more information about how to modify the values of the related parameters, see Set Logtail startup parameters.

Limits on configurations

Item	Description
Configuration update	A custom configuration update requires about 30 seconds to take effect.
Dynamic loading of Logtail configuration files	Supported. The update of a Logtail configuration file does not affect other Logtail configuration files.
Number of Logtail configuration files	Unlimited. However, we recommend that you create a maximum of 100 Logtail configuration files on a server.
Multi-tenant isolation	Logtail configuration files for different tenants are isolated.

Limits on resources and performance metrics

Item	Description
Throughput for log processing	The default traffic of raw logs is limited to 2 MB/s. Data is uploaded after it is encoded and compressed. The compression ratio ranges from 5:1 to 10:1. Logs may be lost if the traffic exceeds the limit. For more information about how to modify the values of the related parameters, see Set Logtail startup parameters.

Item	Description
Maximum processing speed	Single-core processing speed: The maximum processing speed is 100 MB/s for logs in simple mode, 40 MB/s for logs in delimiter mode, and 30 MB/s for logs in JSON mode. By default, the maximum processing speed is 20 MB/s for logs in full regex mode based on the complexity of regular expressions. If multiple processing threads are enabled, the performance can be improved by 1.5 to 3 times.
Number of monitored directories	Logtail limits the depth of monitored directories to reduce the consumption of your resources. If the upper limit is reached, Logtail stops monitoring additional directories and log files. Logtail monitors a maximum of 3,000 directories, including subdirectories.
Number of monitored files	A Logtail configuration file on each server can be used to monitor a maximum of 10,000 files by default. A Logtail client on each server can monitor a maximum of 100,000 files by default. Excessive files are not monitored. If the upper limit is reached, you can perform the following operations: • Improve the depth of the monitored directory in each Logtail configuration file. • Modify the value of the mem_usage_limit parameter to increase the Logtail memory usage threshold. For more information, see Set Logtail startup parameters. You can set a memory usage threshold of 2 GB for Logtail. In this case, the maximum number of files that each Logtail configuration file can be used to monitor is increased to 100,000. The maximum number of files that each Logtail client can monitor is increased to 1,000,000.
Default resources	By default, Logtail consumes up to 40% of CPU usage and 256 MB of memory. If logs are generated at a high speed, you can modify relevant parameters. For more information, see Set Logtail startup parameters.
Processing policy of threshold-crossing resources	If the resources occupied by Logtail exceed the upper limit and this issue lasts for five or more minutes, Logtail is forcibly restarted. The restart may cause data loss or duplication.

Limits on error handling

Item	Description
Network error handling	If a network error occurs, Logtail retries and adjusts the retry interval.
Processing policy of threshold-crossing resources	If the data transmission speed exceeds the quota of the Logstore, Logtail restricts the log collection speed and retries the log collection.
Maximum retry period before timeout	If data fails to be transmitted and the failure lasts for more than six consecutive hours, Logtail discards the data.
Status self-check	Logtail restarts if an exception occurs, for example, an application unexpectedly exits or the resource usage exceeds the quota.

Other limits

Item	Description

Item	Description
Log collection latency	A latency of less than 1 second exists between the time when a log is written to a disk and the time when Logtail collects the log. However, if the log collection speed is restricted, the latency increases.
Log upload policy	Before Logtail uploads logs, it aggregates the logs in the same file. The log upload starts if the number of logs exceeds 2,000, the total size of logs exceeds 2 MB, or the log collection duration exceeds 3 seconds.

28.1.3.2. Other collection methods

28.1.3.2.1. WebTracking

This topic describes how to use WebTracking to collect logs from websites that are written in HTML or HTML5, iOS, and Android.

Context

Log Service uses WebTracking to collect logs from websites written in HTML or HTML5, iOS, and Android. You can customize dimensions and metrics.

In the preceding figure, WebTracking allows you to collect user information from various browsers, iOS apps, and Android apps (except for SDK for iOS or Android). For example:

- Browsers, operating systems, and resolutions used by users.
- Browsing history such as preferences on different websites.
- The length of time that users stay on an app and whether the users are active in the app.

Precautions

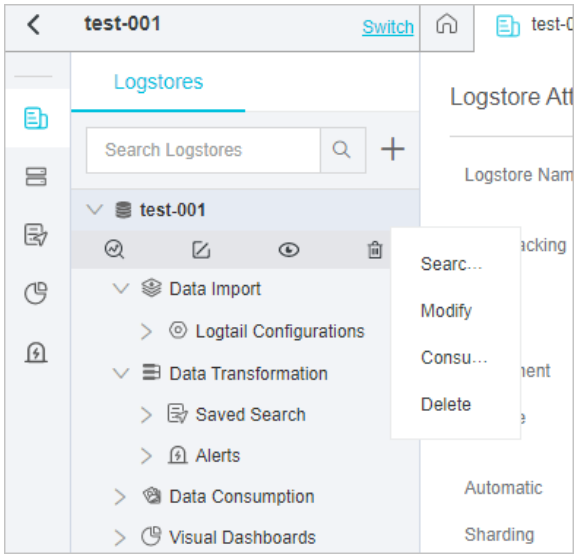
- Dirty data may occur due to the use of WebTracking. This occurs because WebTracking allows unauthorized write access from Internet anonymous users to a Logstore.
- GET and POST requests are supported. The size of a request body cannot exceed 16 KB.
- The same limits apply to POST requests and the PutLogs API operation. The maximum number of logs that you can use a POST request to collect at a time is 4,096. The total size of these logs cannot exceed 3 MB.

Step 1: Enable WebTracking

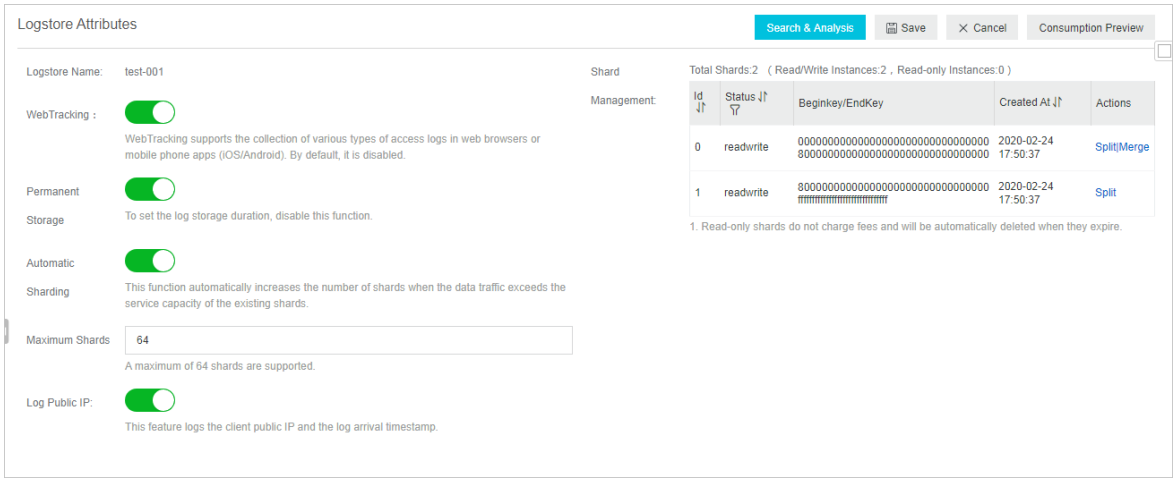
Log on to the Log Service console and enable WebTracking.

1. Log on to the Log Service console.

2. Select the target Logstore, go to the **Logstore Attributes** page.
   - Method 1: Click the Modify icon under the name of the Logstore.
   - Method 1: Click the icon next to the name of the Logstore, and select **Modify**.



3. On the **Logstore Attributes** page, click **Modify** in the upper-right corner.
4. Turn on the WebTracking switch.



Use SDK for Java to enable WebTracking

Example:

```
import com.aliyun.openservices.log.Client;
import com.aliyun.openservices.log.common.LogStore;
import com.aliyun.openservices.log.exception.LogException;
public class WebTracking {
  static private String accessId = "your accesskey id";
  static private String accessKey = "your accesskey";
  static private String project = "your project";
  static private String host = "log service data address";
  static private String logStore = "your logstore";
  static private Client client = new Client(host, accessId, accessKey);
  public static void main(String[] args) {
      try {
          //Enable WebTracking on an existing Logstore.
          LogStore logSt = client.GetLogStore(project, logStore).GetLogStore();
          client.UpdateLogStore(project, new LogStore(logStore, logSt.GetTtl(), logSt.GetShardCount()
, true));
          //Disable WebTracking.
          //client.UpdateLogStore(project, new LogStore(logStore, logSt.GetTtl(), logSt.GetShardCount
(), false));
          //Create a Logstore on which you want to enable WebTracking.
          //client.UpdateLogStore(project, new LogStore(logStore, 1, 1, true));
      }
      catch(LogException e){
          e.printStackTrace();
      }
  }
}
```

Step 2: Collect logs

After you enable WebTracking for a Logstore, you can use the following methods to upload data to the Logstore.

- Use the JavaScript SDK
  i. Copy the *loghub-tracking.js* file to the *web* directory and add the following script to the file.

     Click here to copy the script.

     ```
     <script type="text/javascript" src="loghub-tracking.js" async></script>
     ```

     > **Note** To ensure that a page loads, the script asynchronously sends HTTP requests. If data must be sent several times during the page loading, the newest request overwrites the previous HTTP request. A message showing WebTracking is about to exit appears in the browser. To eliminate the issue, send requests in a synchronous manner. To implement the method, perform the following step.
     >
     > Original statement:
     >
     > ```
     > this.httpRequest_.open("GET", url, true)
     > ```
     >
     > Replace the original statement with the following statement:
     >
     > ```
     > this.httpRequest_.open("GET", url, false)
     > ```

  ii. Create a tracker.

```
var logger = new window.Tracker('${host}','${project}','${logstore}');
logger.push('customer', 'zhangsan');
logger.push('product', 'iphone 6s');
logger.push('price', 5500);
logger.logger();
logger.push('customer', 'lisi');
logger.push('product', 'ipod');
logger.push('price', 3000);
logger.logger();
```

The following table lists the parameters:

Parameter	Description
`${host}`	The endpoint of the region where Log Service resides. For more information, see the *Obtain an endpoint topic in the Log Service Developer Guide* .
`${project}`	The name of the project that you create in Log Service.
`${logstore}`	The name of the Logstore in the `${project}` .

After you run the following code, the following logs appear in Log Service.

```
customer:zhangsan
product:iphone 6s
price:5500
```

```
customer:lisi
product:ipod
price:3000
```

- Use HTTP GET requests

```
curl --request GET 'http://${project}.${host}/logstores/${logstore}/track? APIVersion=0.6.0&key1=val1&key2=val2'
```

The following table lists the parameters.

Parameter	Description
${project}	The name of the project that you create in Log Service.
${host}	The endpoint of the region where Log Service resides.
${logstore}	The name of a Logstore that has WebTracking enabled in the `${project}` .
APIVersion=0.6.0	(Required) A reserved parameter.
__topic__=yourtopic	(Optional) A reserved parameter that specifies the topic of the log.
key1=val1, key2=val2	The key-value pairs that you want to upload to Log Service. You can specify multiple pairs. Make sure that the length of each request URL is less than 16 KB.

- Use HTML img tags

```
<img src='http://${project}.${host}/logstores/${logstore}/track.gif? APIVersion=0.6.0&key1=val1&key2=val2'/>
<img src='http://${project}.${host}/logstores/${logstore}/track_ua.gif? APIVersion=0.6.0&key1=val1&key2=val2'/>
```

The parameters that you need to specify are the same as the preceding parameters. In addition to custom parameters that are uploaded by track_ua.gif, Log Service also uses UserAgent and referer fields in the HTTP header as the fields of logs.

28.1.3.2.2. Use SDKs to collect logs

28.1.3.2.2.1. Producer Library

The Aliyun LOG Java Producer supports Java applications that run in big data processing scenarios with high concurrency. The library is easy to use and highly customizable.

For more information about the related GitHub project, visit Aliyun LOG Java Producer.

28.1.3.2.2.2. Log4j Appender

Log4j is an open-source logging framework of Apache. You can use Log4j to write logs to the Log Service console, files, graphical user interface (GUI) components, socket servers, NT kernel loggers, or Unix syslog daemons. You can specify the output format of each log. You can also specify the severity level of each log to implement a fine-grained control on log generation.

Log4j consists of three components: loggers, appenders, and layouts.

- Loggers allow you to specify the severity level of each log.

  Severity levels are sorted into ERROR, WARN, INFO, and DEBUG in descending order of severity.
- Appenders allow you to specify the destination of each log.

  A destination can be the Log Service console or a file.
- Layouts allow you to specify the output format of each log.

  The output format defines how logs are displayed.

To write logs to Log Service, use the Alibaba Cloud Log Log4j Appender. For information about where to download the library and how to use it, see Log4j Appender.

28.1.3.2.2.3. Logback Appender

Logback was created by the same developer of Log4j. Logback allows you to write logs to multiple destinations. These destinations include the Log Service console, files, graphical user interface (GUI) components, socket servers, NT kernel loggers, and Unix syslog daemons. You can define the output format of each log. If you define the severity level of each log, you can implement a fine-grained control on the log generation process.

You can set the destination of logs to Log Service by using the Aliyun Log Logback Appender. The following example shows the format of logs that are uploaded to Log Service.

```
level: ERROR
location: com.aliyun.openservices.log.logback.example.LogbackAppenderExample.main(LogbackAppenderExam
ple.java:18)
message: error log
throwable: java.lang.RuntimeException: xxx
thread: main
time: 2018-01-02T03:15+0000
log: 2018-01-02 11:15:29,682 ERROR [main] com.aliyun.openservices.log.logback.example.LogbackAppender
Example: error log
__source__: xxx
__topic__: yyy
```

For information about where to download the library and how to use it, see Logback Appender.

28.1.3.2.2.4. Golang Producer Library

The Aliyun LOG Go Producer Library supports Go applications that run in big data processing scenarios with high concurrency. The library is easy to use and highly customizable. You can use the library to create producers that allow you to resend failed logs. Before Go applications send log data to Log Service, you can use these producers to compress the log data. This improves write performance.

For more information about the related GitHub project, visit Aliyun Log Go Producer.

28.1.3.2.2.5. Python logging

Configurations

For more information about the configurations that are related to the Python logging module, see Logging configuration.

The Python logging module allows you to configure logging by using code or a configuration file. The following example shows how to configure logging by using the `logging.conf` configuration file.

```
[loggers]
keys=root,sls
[handlers]
keys=consoleHandler, slsHandler
[formatters]
keys=simpleFormatter, rawFormatter
[logger_root]
level=DEBUG
handlers=consoleHandler
[logger_sls]
level=INFO
handlers=consoleHandler, slsHandler
qualname=sls
propagate=0
[handler_consoleHandler]
class=StreamHandler
level=DEBUG
formatter=simpleFormatter
args=(sys.stdout,)
[handler_slsHandler]
class=aliyun.log.QueuedLogHandler
level=INFO
formatter=rawFormatter
args=(os.environ.get('ALIYUN_LOG_SAMPLE_ENDPOINT', ''), os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSID',
''), os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSKEY', ''), os.environ.get('ALIYUN_LOG_SAMPLE_TMP_PROJECT
', ''), "logstore")
[formatter_simpleFormatter]
format=%(asctime)s - %(name)s - %(levelname)s - %(message)s
[formatter_rawFormatter]
format=%(message)s
```

Two handlers named `root` and `sls` are created. The `sls` handler is an object of the `aliyun.log.QueuedLogHandler` class. The following shows the parameters that are specified for the sls handler. For more information, see Parameters.

```
args=(os.environ.get('ALIYUN_LOG_SAMPLE_ENDPOINT', ''), os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSID',
''), os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSKEY', ''), os.environ.get('ALIYUN_LOG_SAMPLE_TMP_PROJECT
', ''), "logstore")
```

> **Note** In this case, the `os.environ` function is used to retrieve configurations from environment variables. You can also specify values for these parameters based on your business requirements.

Upload logs

You can use the configuration file to upload logs to Log Service.

```
import logging
import logging.config
# Configurations
logging.config.fileConfig('logging.conf')
logger = logging.getLogger('sls')
# Use the logger
logger.info("test1")
try:
    1/0
except ZeroDivisionError as ex:
    logger.exception(ex)
```

Then, logs are automatically uploaded to Log Service. To use the LogSearch/Analytics feature, you must enable the index feature on the corresponding Logstore.

Configure an index for a Logstore

Enable the index feature on the Logstore that receives logs and configure an index for specific fields. We recommend that you use CLI (Command Line Interface) to configure the index as follows:

```
aliyunlog log update_index --project_name="project1" --logstore_name="logstore1" --index_detail="file
:///Users/user1/loghandler_index.json"
```

For more information, see the python_logging_handler_index.json configuration file.

Specify log fields to be collected

The following table lists supported log fields that you can collect. By default, all of the fields are collected.

Field	Description
message	The contents of a log.
record_name	The name of a handler. In the preceding example, the name is `sls`.
level	The output level of a logger, such as INFO and ERROR.
file_path	The full path of a configuration file.
func_name	The name of a function.
line_no	The number of a log line.
module	The name of a module where the function resides.
thread_id	The ID of the thread that runs the function.
thread_name	The name of the thread that runs the function.
process_id	The ID of the process that runs the function.
process_name	The name of the process that runs the function.

You can specify log fields to be collected based on the fields parameter of the `QueuedLogHandler` class. For more information, see aliyun.log.LogFields.

The following example shows how to modify the preceding configuration file and collect several fields, such as module and func_name.

```
[handler_slsHandler]
class=aliyun.log.QueuedLogHandler
level=INFO
formatter=rawFormatter
args=('cn-beijing.log.aliyuncs.com', 'ak_id', 'ak_key', 'project1', "logstore1", 'mytopic', ['level',
'func_name', 'module', 'line_no']  )
```

Note

- The message field is collected regardless of your configurations.
- To add a prefix and suffix to the names of these fields, use the `buildin_fields_prefix` and `buildin_fields_suffix` parameters. For example, `__level__`.

Configure logging by using a JSON text

You can use a JSON text to create more flexible logging configurations than code does.

```
#encoding: utf8
import logging, logging.config, os
# Configurations
conf = {'version': 1,
        'formatters': {'rawformatter': {'class': 'logging.Formatter',
                                        'format': '%(message)s'}
                       },
        'handlers': {'sls_handler': {'()':
                                     'aliyun.log.QueuedLogHandler',
                                     'level': 'INFO',
                                     'formatter': 'rawformatter',
                                     # custom args:
                                     'end_point': os.environ.get('ALIYUN_LOG_SAMPLE_ENDPOINT', ''),
                                     'access_key_id': os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSID', ''
),
                                     'access_key': os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSKEY', ''),
                                     'project': 'project1',
                                     'log_store': "logstore1"
                                     }
                     },
        'loggers': {'sls': {'handlers': ['sls_handler', ],
                                    'level': 'INFO',
                                    'propagate': False}
                    }
        }
logging.config.dictConfig(conf)
# Use the logger
logger = logging.getLogger('sls')
logger.info("Hello world")
```

Note To instantiate an object of the `aliyun.log.QueuedLogHandler` class, pass named parameters to the constructor. For more information, see Parameters.

28.1.3.2.3. Common log formats

28.1.3.2.3.1. Log4j logs

Log Service allows you to collect Log4j logs.

Collect Log4j logs by using LogHub Log4j Appender

For more information, see Log4j Appender.

Configure Logtail to collect Log4j logs

This topic describes how to configure regular expressions based on the default configuration of Log4j 1 logs. If Log4j 2 is used, you must modify the default configuration to record the complete date information. Log4j logs are sorted into Log4j 1 logs and Log4j 2 logs.

```
<Configuration status="WARN">
  <Appenders>
    <Console name="Console" target="SYSTEM_OUT">
      <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss:SSS zzz} [%t] %-5level %logger{36} - %msg%n"/>
    </Console>
  </Appenders>
  <Loggers>
    <Logger name="com.foo.Bar" level="trace">
      <AppenderRef ref="Console"/>
    </Logger>
    <Root level="error">
      <AppenderRef ref="Console"/>
    </Root>
  </Loggers>
</Configuration>
```

For more information about how to configure Logtail to collect Log4j logs, see Python logs. Configure the required parameters based on your network environment and business requirements.

The automatically generated regular expression is based on the sample log and may not be suitable for other logs. Therefore, you must make minor changes to the regular expression after it is automatically generated

The following shows a sample log of the default Log4j format.

```
2013-12-25 19:57:06,954 [10.207.37.161] WARN impl.PermanentTairDaoImpl - Fail to Read Permanent Tair, key:e:470217319319741_1,result:com.example.tair.Result@172e3ebc[rc=code=-1, msg=connection error or timeout,value=,flag=0]
```

Regular expression that matches IP addresses that each indicate the start of a line:

```
\d+-\d+-\d+\s. *
```

Regular expression used to extract log information:

```
(\d+-\d+-\d+\s\d+:\d+:\d+,\d+)\s\[([^\]]*)\]\s(\S+)\s+(\S+)\s-\s(. *)
```

Time conversion format:

```
%Y-%m-%d %H:%M:%S
```

The following table lists the extraction results of the sample log.

Key	Value
time	2013-12-25 19:57:06,954

Key	Value
ip	10.207.37.161
level	WARN
class	impl.PermanentTairDaoImpl
message	Fail to Read Permanent Tair,key:e:470217319319741_1,result:com.example.tair.Result@172e3ebc[rc=code=-1, msg=connection error or timeout,value=,flag=0]

28.1.3.2.3.2. Python logs

The Python logging module provides a general logging system, which can be used by third-party modules or applications.

The Python logging module provides different log levels and logging methods, such as file-based, HTTP GET, HTTP POST, SMTP, and Socket logging. You can also create a custom logging method. The Python logging module works in the same way as the Log4j logging module except for some implementation details. The Python logging module includes the logger, handler, filter, and formatter objects.

Log format

A formatter specifies the output format of logs. To instantiate a formatter, pass two parameters to the constructor. One parameter includes a message format string and the other parameter includes a date format string. The parameters are optional.

Log format:

```
import logging
import logging.handlers
LOG_FILE = 'tst.log'
handler = logging.handlers.RotatingFileHandler(LOG_FILE, maxBytes = 1024*1024, backupCount = 5) # Ins
tantiate the handler
fmt = '%(asctime)s - %(filename)s:%(lineno)s - %(name)s - %(message)s'
formatter = logging.Formatter(fmt)   # Instantiate the formatter
handler.setFormatter(formatter)      # Add the formatter to the handler
logger = logging.getLogger('tst')    # Obtain a logger named tst
logger.addHandler(handler)           # Add the handler to the logger
logger.setLevel(logging.DEBUG)
logger.info('first info message')
logger.debug('first debug message')
```

Attributes

Formatter attributes are specified in the `%(key)s` format. The following table lists the attributes.

Format	Description
%(name)s	The name of a logger that generates logs.
%(levelno)s	The log output level in the numeric format. Valid values: DEBUG, INFO, WARNING, ERROR, and CRITICAL
%(levelname)s	The log output level in the text format. Valid values: 'DEBUG', 'INFO', 'WARNING', 'ERROR', and 'CRITICAL'.

Format	Description
%(pathname)s	The full path of a source file that contains the logging module.
%(filename)s	The name of the source file.
%(module)s	The name of a module where the statement that you use to generate logs resides.
%(funcName)s	The name of the function that is used to call the log output function.
%(lineno)d	The number of a code line that contains the statement used to call the log output function.
%(created)f	The time when the log was created. The value is a UNIX timestamp representing the number of seconds that have elapsed since January 1, 1970, 00:00:00 (UTC).
%(relativeCreated)d	The interval between the time when a log was created and the time when the logging module was loaded. Unit: milliseconds.
%(asctime)s	The time when the log was created. The value of 2003-07-08 16:49:45,896 is an example of the default format. The number after the comma (,) indicates the number of milliseconds.
%(msecs)d	The time when the log was created. The value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 (UTC).
%(thread)d	The thread ID.
%(threadName)s	The thread name.
%(process)d	The process ID.
%(message)s	The contents of a log.

Sample logs

Sample log:

```
2015-03-04 23:21:59,682 - log_test.py:16 - tst - first info message
2015-03-04 23:21:59,682 - log_test.py:17 - tst - first debug message
```

Common Python logs and the corresponding regular expressions:

- Sample log:

  ```
  2016-02-19 11:03:13,410 - test.py:19 - tst - first debug message
  ```

  Regular expression:

  ```
  (\d+-\d+-\d+\s\S+)\s+-\s+([^:]+):(\d+)\s+-\s+(\w+)\s+-\s+(. *)
  ```

- Log format

  ```
  %(asctime)s - %(filename)s:%(lineno)s - %(levelno)s %(levelname)s %(pathname)s %(module)s %(funcName)s %(created)f %(thread)d %(threadName)s %(process)d %(name)s - %(message)s
  ```

  Sample log:

```
2016-02-19 11:06:52,514 - test.py:19 - 10 DEBUG test.py test <module> 1455851212.514271 1398659966
87072 MainThread 20193 tst - first debug message
```

Regular expression:

```
(\d+-\d+-\d+\s\S+)\s-\s([^:]+):(\d+)\s+-\s+(\d+)\s+(\w+)\s+(\S+)\s+(\w+)\s+(\S+)\s+(\S+)\s+(\d+)\s
+(\w+)\s+(\d+)\s+(\w+)\s+-\s+(. *)
```

Configure Logtail to collect Python logs

1. Log on to the Log Service console.
2. Click **Import Data**. On the **Import Data** page that appears, select **RegEx-Text Log**.
3. Select a destination project and Logstore, and then click **Next**.

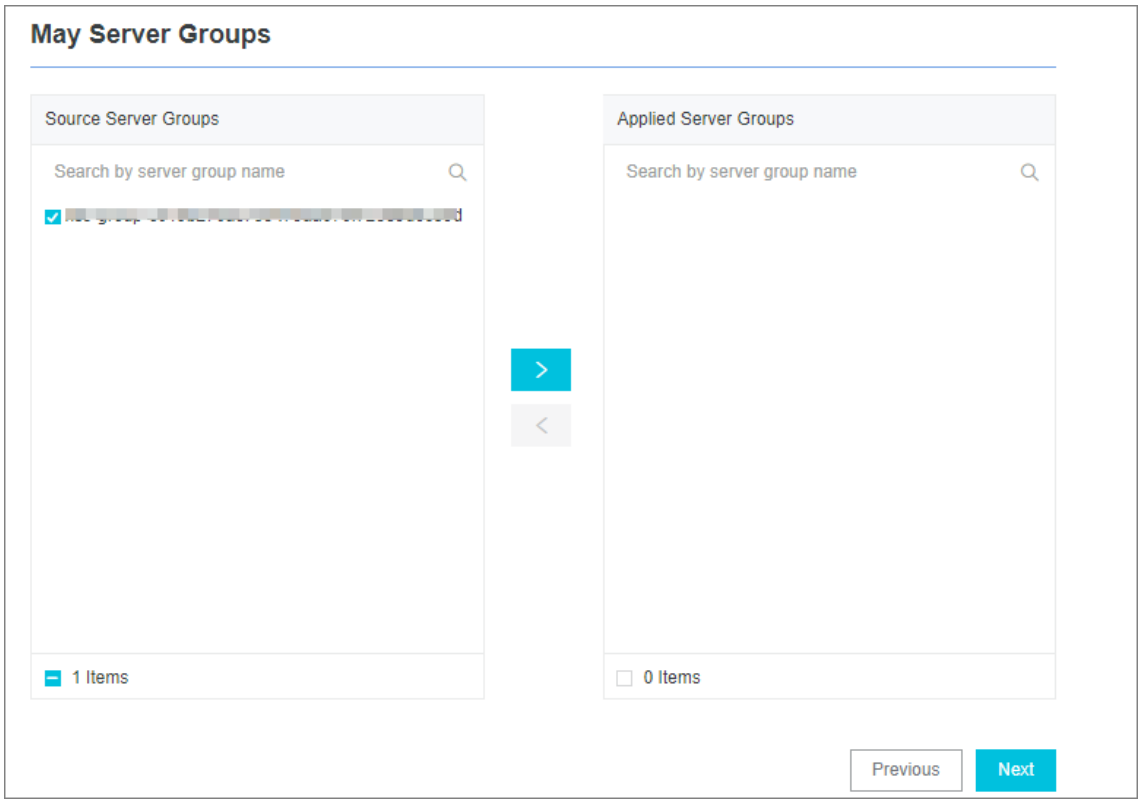
   You can also click **Create Now** to create a project and a Logstore.

   If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.
4. Create a machine group and click **Next**.

   Before you can create a machine group, you must install Logtail.

   Install Logtail as prompted. For more information, see Install Logtail in Linux or Install Logtail in Windows.

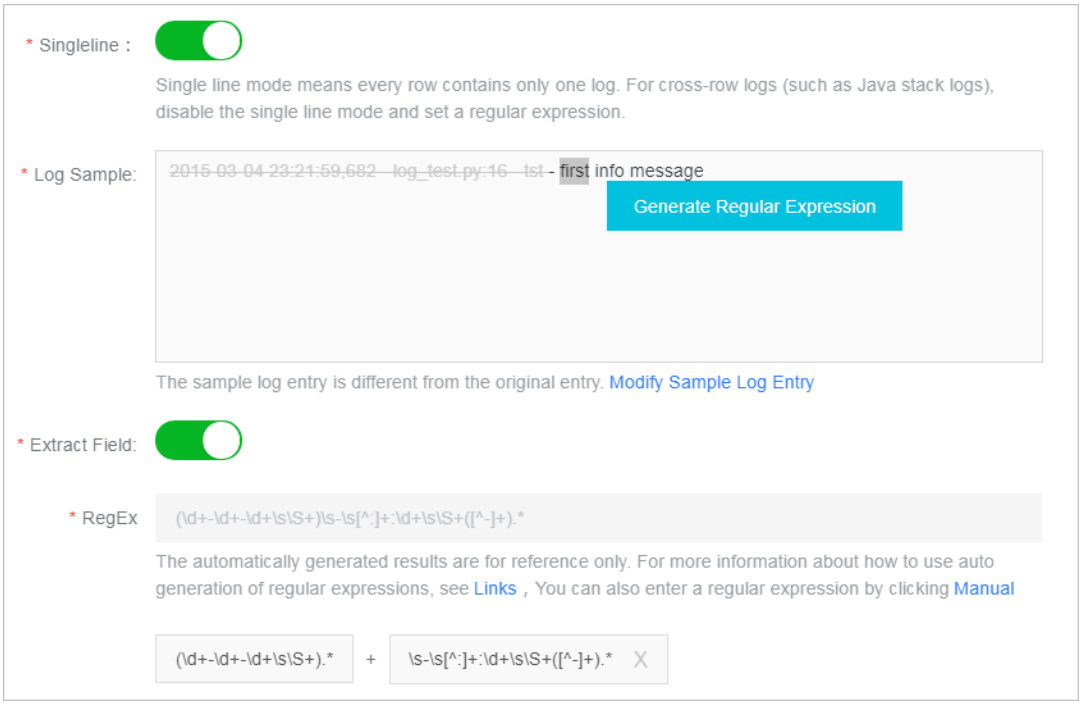
   After you install Logtail, click **Complete Installation** to create a machine group. If a machine group is created, click **Use Existing Machine Groups** to select the machine group.
5. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



Notice If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

6. Create a Logtail configuration.
   i. Enter a **configuration name** and **log path** and select **Full Regex Mode** in the Mode field.
   ii. Turn on **Singleline**.
   iii. Enter a snippet in the **Log Sample** field.
   iv. Turn on **Extract Field**.
   v. Set a regular expression in the **RegEx** field.
      a. Select fields to generate a regular expression

         If the regular expression that is automatically generated does not match your sample log, you can select fields in the sample log to generate a regular expression. Log Service can automatically parse the highlighted fields of the sample log to generate a regular expression. In the **Log Sample** field, select the required fields, and click **Generate Regular Expression**. The regular expression of the selected field is displayed in the **RegEx** field. To obtain a full regular expression for the sample log, generate regular expressions for each log field.



      b. Modify the regular expression

         Actual data formats may vary. In this case, click **Manual** under the RegEx field to adjust the regular expression that is automatically generated based on your business requirements. This ensures that the regular expression is suitable for all formats of the collected logs.

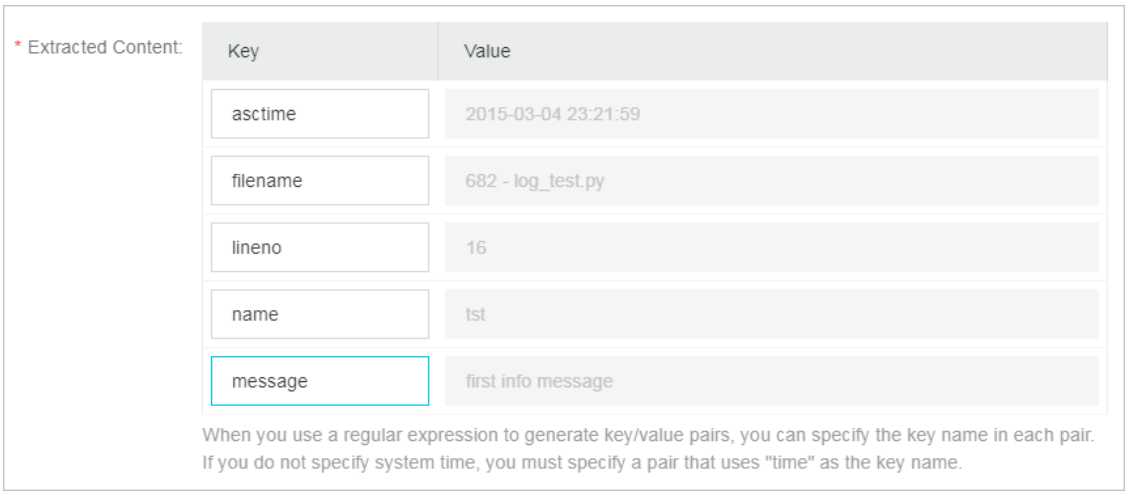
      c. Verify the regular expression

         After you modify the regular expression, click **Validate** next to the RegEx field. If the regular expression is valid, the extraction results are displayed. If the regular expression is invalid, modify the regular expression again.

vi. Confirm **the extraction results of log fields**.

View the extraction results of log fields and specify keys for the extracted fields.

Specify an informative name for each log field in the extraction results. For example, time as the name for a time field. If you do not use the system time, you must specify the name of a time field in the Value fields and time in the Key field.



7. (Optional)Specify **Advanced Options** and click **Next**.

Set the parameters in the Advanced Options section based on your business requirements. We recommend that you do not modify the settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs. **Note** If you turn on **Enable Plug-in Processing**, specific parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.
Upload Raw Log	If you turn on **Upload Raw Log**, each raw log is uploaded to Log Service as a value of the __raw__ field together with the parsed log.
Topic Generation Mode	The topic generation mode. ○ **Null - Do not generate topic**: This mode is selected by default. In this mode, the topic field is set to an empty string. You do not need to enter a topic to query logs. ○ **Machine Group Topic Attributes**: This mode is used to differentiate logs that are generated by different servers. ○ **File Path RegEx**: In this mode, you must specify a regular expression in the **Custom RegEx** field. The part of a log path that matches the regular expression is used as the topic. This mode is used to differentiate logs that are generated by different users or instances.
Custom RegEx	If you set the Topic Generation Mode parameter to **File Path RegEx**, you must enter a custom regular expression.

Parameter	Description
Log File Encoding	The encoding format of log files. Valid values: ○ utf8: UTF-8 encoding format ○ gbk: GBK encoding format
Timezone	The time zone where logs are collected. Valid values: ○ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ○ Custom: If you select this value, you must select a time zone.
Timeout	The timeout period of log files. If a log file is not updated within the specified period, Logtail reckons the file to be timed out. Valid values: ○ Never: All log files are continuously monitored and never time out. ○ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select **30 Minute Timeout**, you must specify the **Maximum Timeout Directory Depth** parameter. Valid values: 1 to 3.
Filter Configuration	Only logs that **meet all filter conditions** are collected. Examples: ○ Collect logs that meet specified conditions: If you set **Key** to **level** and **Regex** to **WARNING|ERROR**, only WARNING-level and ERROR-level logs are collected. ○ Filter out logs that do not meet specified conditions. ■ If you set **Key** to **level** and **Regex** to **^(?!.*(INFO|DEBUG)).***, INFO-level or DEBUG-level logs are not collected. ■ If you set **Key** to **url** and **Regex** to **.*^(?!.*(healthcheck)).***, logs whose URLs contain healthcheck are not collected. For example, the log is not collected if the value of the Key field is url and the value of the Value field is /inner/healthcheck/jiankong.html.

8. Configure indexes in the Configure Query and Analysis step. Click **Next**.

   By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Enable the index feature and configure indexes for a Logstore.

   > **Note**
   > ○ To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings of Field Search prevail.
   > ○ If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

After the configuration is complete, apply the settings to the server group to collect Python logs.

28.1.3.2.3.3. Node.js logs

Node.js logs are displayed in the Log Service console by default. This impedes data collection and troubleshooting. You can use the log4js function to write logs into files and customize the log format. This facilitates data collection and consolidation.

Example:

```
var log4js = require('log4js');
log4js.configure({
  appenders: [
    {
      type: 'file', //Output to a file
      filename: 'logs/access.log',
      maxLogSize: 1024,
      backups:3,
      category: 'normal'
    }
  ]
});
var logger = log4js.getLogger('normal');
logger.setLevel('INFO');
logger.info("this is a info msg");
logger.error("this is a err msg");
```

Log format

After logs are written to text files by using the log4js function, these logs are displayed in the following format:

```
[2016-02-24 17:42:38.946] [INFO] normal - this is a info msg
[2016-02-24 17:42:38.951] [ERROR] normal - this is a err msg
```

The log4js function defines six log severity levels. They are TRACE, DEBUG, INFO, WARN, ERROR, and FATAL in ascending order of severity.

Use Logtail to collect Node.js logs

For more information about how to configure Logtail to collect Python logs, see Python logs. Use configurations based on your network environment and business requirements.

The automatically generated regular expression is based on the sample log and may not apply to other logs. Therefore, you must make minor changes to the regular expression after it is generated. You can use the following sample Node.js logs to configure appropriate regular expressions for your logs.

Sample Node.js logs:

- Example 1
  - Sample log

    ```
    [2016-02-24 17:42:38.946] [INFO] normal - this is a info msg
    ```

  - Regular expression:

    ```
    \[([^]]+)]\s\[([^\]]+)]\s(\w+)\s-(. *)
    ```

  - Extracted fields:

    `time` , `level` , `loggerName` , and `message`

- Example 2
  - Sample log

    ```
    [2016-01-31 12:02:25.844] [INFO] access - 42.120.73.203 - - "GET /user/projects/ali_sls_log? ign
    oreError=true HTTP/1.1" 304 - "http://
    aliyun.com/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_3) AppleWebKit/537.36 (KHTML, like Ge
    cko) Chrome/48.0.2564.97 Safari/537.36"
    ```

- Regular expression:

```
\[([^]]+)]\s\[(\w+)]\s(\w+)\s-\s(\S+)\s-\s-\s"([^"]+)"\s(\d+)[^"]+("[^"]+)"\s"([^"]+). *
```

- Extracted fields:

`time` , `level` , `loggerName` , `ip` , `request` , `status` , `referer` , and `user_agent`

28.1.3.2.3.4. WordPress logs

This topic describes the format of WordPress logs and extraction results of a sample log.

Log format

Sample log:

```
172.64.0.2 - - [07/Jan/2016:21:06:39 +0800] "GET /wp-admin/js/password-strength-meter.min.js? ver=4.4 HTTP/1.0" 200 776 "http://wordpress.c4a1a0aecdb1943169555231dcc4adfb7.cn-hangzhou.alicontainer.com/wp-admin/install.php" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36"
```

Configure Logtail to collect WordPress logs

Configurations required to collect WordPress logs:

- Regular expression that matches IP addresses that each indicate the start of a line

```
\d+\.\d+\.\d+\.\d+\s-\s. *
```

- Regular expression used to extract information from the log:

```
(\S+) - - \[([^\]]*)] "(\S+) ([^"]+)" (\S+) (\S+) "([^"]+)" "([^"]+)"
```

- Time conversion format:

```
%d/%b/%Y:%H:%M:%S
```

- Results after Logtail extracts information from the sample log

Key	Value
ip	10.10.10.1
time	07/Jan/2016:21:06:39 +0800
method	GET
url	/wp-admin/js/password-strength-meter.min.js? ver=4.4 HTTP/1.0
status	200
length	776
ref	http://wordpress.c4a1a0aecdb1943169555231dcc4adfb7.cn-hangzhou.alicontainer.com/wp-admin/install.php
user-agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36

28.1.3.2.3.5. Unity3D logs

Log Service can use the WebTracking feature to collect Unity3D logs. The following example shows how to collect *Unity logs* of the debug type.

Context

Unity3D is a cross-platform game engine developed by Unity Technologies. The engine allows you to create 3D video games, VR buildings, real-time 3D animation, and other interactive content.

Procedure

1. Enable the WebTracking feature.

   For more information about how to enable this feature, see WebTracking.

2. Create a Unity3D LogHandler.

   In a Unity editor, create a C# file named *LogOutputHandler.cs*, add the following code, and modify the following variables.

   - project: specifies the name of the project.
   - logstore: specifies the name of the Logstore.
   - serviceAddr: specifies the endpoint of the project.

     For more information about serviceAddr, see View the information of a project.

```
using UnityEngine;
using System.Collections;
public class LogOutputHandler : MonoBehaviour
{
    //Register the HandleLog function on scene start to fire on debug.log events
    public void OnEnable()
    {
        Application.logMessageReceived += HandleLog;
    }
    //Remove callback when object goes out of scope
    public void OnDisable()
    {
        Application.logMessageReceived -= HandleLog;
    }
    string project = "your project name";
    string logstore = "your logstore name";
    string serviceAddr = "http address of your log service project";
    //Capture debug.log output, send logs to Loggly
    public void HandleLog(string logString, string stackTrace, LogType type)
    {
        string parameters = "";
        parameters += "Level=" + WWW.EscapeURL(type.ToString());
        parameters += "&";
        parameters += "Message=" + WWW.EscapeURL(logString);
        parameters += "&";
        parameters += "Stack_Trace=" + WWW.EscapeURL(stackTrace);
        parameters += "&";
        //Add any User, Game, or Device MetaData that would be useful to finding issues later
        parameters += "Device_Model=" + WWW.EscapeURL(SystemInfo.deviceModel);
        string url = "http://" + project + "." + serviceAddr + "/logstores/" + logstore + "/track
? APIVersion=0.6.0&" + parameters;
        StartCoroutine(SendData(url));
    }
    public IEnumerator SendData(string url)
    {
        WWW sendLog = new WWW(url);
        yield return sendLog;
    }
}
```

The preceding code allows you to send logs to Log Service in an asynchronous manner. In the code, you can specify more fields you want to collect.

3. Generate Unity logs.

   In the project, create a C# file named *LogglyTest.cs* and add the following code.

```
using UnityEngine;
using System.Collections.Generic;
public class LogglyTest : MonoBehaviour {
    void Start () {
        Debug.Log ("Hello world");
    }
}
```

4. View logs in the console.

   After you complete the preceding steps, run the Unity application. In the Log Service console, view logs that are sent to Log Service.

The preceding code shows how to use *Debug.Log*, *Debug.LogError*, and *Debug.LogException* methods to collect logs. Unity provides Component Object Model (COM)-based exception handling and log handling APIs. These APIs allow you to easily collect device details of clients.

28.1.4. Query and analysis

28.1.4.1. Overview

Log Service provides the LogSearch/Analytics feature that you can use to query and analyze a large number of logs. If you do not enable indexes, raw data is consumed in sequence based on shards. The procedure is similar to the sequential consumption of Kafka messages. If you enable indexes, you can query logs and perform statistical analysis on query results in addition to consuming logs in sequence.

Benefits

- Real-time: Logs can be analyzed immediately after they are written.
- Fast:
  - Query: Billions of data records can be processed and queried within one second. Each search statement has a maximum of five conditions specified.
  - Analysis: Hundreds of millions of data records can be aggregated and analyzed within one second. Each query has a maximum of five aggregate functions and a GROUP BY clause specified.
- Flexible: Query and analysis conditions can be changed as required and the results are returned in real time.
- All-in-one: Reports and dashboards are available in the console for quick analysis. In addition to these features, Log Service can work together with Grafana, DataV, Jaeger, and other services. It also supports RESTful APIs, Java Database Connectivity (JDBC) APIs, and other APIs.

Indexing

Indexes refer to a data structure that you can use to sort the values of one or more columns of logs. Indexes allow you to obtain the required information in a timely manner from logs that Log Service collects. Before you use the LogSearch/Analytics feature, you must collect logs and Enable the index feature and configure indexes for a Logstore on the collected logs.

In Log Service, indexes are sorted into **full-text indexes** and **field-specific indexes**.

- **Full-text index**: Indexing is enabled for the full contents of a log. The values of all fields in a log are queried by default. The log can be queried if one of the fields matches the search term.
- **Field-specific index**: You can configure a field-specific index for a key. Then, you can query logs based on specific keys to narrow the query scope.

To use **field-specific indexes**, you must specify the data type for a field. Available data types for fields in Log Service include Text, JSON, Long, and Double. For more information about Overview.

Query methods

- Console

  In the Log Service console, you can query logs by specifying time ranges and search statements. For more information about the procedure and search statements, see Query logs and Query syntax.

- API

  To query logs, you can call the GetLogs and GetHistograms API operations of the Log Service API.

> **Note** Before you query logs, make sure that you collect logs and Enable the index feature and configure indexes for a Logstore.

Search and analysis statements

To apply real-time LogSearch/Analytics to collected logs, you must specify query statements. Each query statement includes the search section and the analytics section. Separate the sections with a vertical bar (|).

```
$Search |$Analytics
```

Statement	Required	Description
Search	No	A search statement contains search conditions. These conditions include keywords, fuzzy keywords, values, ranges, and combined conditions. If you leave the statement empty or specify an asterisk (*) for the statement, it indicates that no condition is specified and all data is returned. For more information, see Query syntax.
Analytics	No	You can use an analytics statement to aggregate or analyze data based on query results. If you leave the statement empty, it indicates that no analytics is required and all query results are returned. For more information, see Real-time analysis.

Precautions

You may query a large number of logs. For example, if the number of logs to be queried is more than 1,000,000,000, Log Service may fail to return all results. Log Service returns a partial result set and notifies you that the returned data set includes partial results.

Query results are cached every 15 minutes. If a partial result set is matched in the cache, Log Service continues to scan logs that are not cached. Log Service combines query results of the current query with results of cached results.

Therefore, Log Service enables you to obtain results by calling the API operation multiple times with the same parameters.

28.1.4.2. Real-time analysis

Log Service supports SQL-like aggregate calculation. This feature integrates search statements with SQL aggregate functions to calculate query results.

Sample statement:

```
status>200 |select avg(latency),max(latency) ,count(1) as c GROUP BY  method  ORDER BY c DESC  LIMIT 20
```

Basic syntax:

```
[search query] | [sql query]
```

Separate a search statement and a calculation statement with a vertical bar (|). You can use the search statement to query logs and obtain the required results. Then, use the calculation statement for further aggregation. The search query syntax is specific to Log Service. For more information, see Query syntax.

Prerequisites

To use the statistical analysis feature, click **Index Attributes**. Turn on the **Enable Analytics** switch for the required field. For more information, see Enable the index feature and configure indexes for a Logstore.

- If you turn off the switch, Log Service calculates up to 10,000 rows of data in each shard along with high latency of calculation.
- If you turn on the switch, Log Service analyzes data in seconds.
- The update applies only to new data.
- No extra fee is incurred for the update.

Supported SQL syntax

Log Service supports the following SQL syntax. For more information about a specific topic, click the corresponding link.

- Aggregate functions that are available for SELECT statements include:
  - General aggregate functions
  - Security check functions
  - Map functions
  - Approximate functions
  - Mathematical statistics functions
  - Mathematical calculation functions
  - String functions
  - Date and time functions
  - URL functions
  - Regular expression functions
  - JSON functions
  - Type conversion functions
  - IP functions
  - Array functions
  - Binary string functions
  - Bitwise operations
  - Interval-valued comparison and periodicity-valued comparison functions
  - Comparison functions and operators
  - Lambda functions
  - Logical functions
  - Geospatial functions
  - Geography functions
  - Machine learning functions
- GROUP BY syntax
- Window functions
- HAVING syntax
- ORDER BY syntax
- LIMIT syntax
- Syntax for CASE statements and if() functions
- UNNEST function
- Field aliases
- Nested subqueries

Precautions

Before you use SQL statements, note the following items:

- You do not need to specify FROM and WHERE clauses for SQL statements. By default, Log Service queries logs from the current Logstore, and each WHERE clause is specified in the [search query] section.
- The supported clauses include SELECT, GROUP BY, ORDER BY [ASC,DESC], LIMIT, and HAVING.

> **Note** By default, only the first 10 results are returned. If you want to return more results, add a LIMIT clause to the statement. For example, `* | select count(1) as c, ip group by ip order by c desc limit 100`.

Built-in fields

Log Service has multiple built-in fields for statistical analysis. A built-in field is automatically added to a valid column that you create.

Field	Type	Description
`__time__`	Bigint	The time when a log was created.
`__source__`	Varchar	The source IP address of a log. When you query logs, the name of the field is source. If you specify the field for an SQL statement, you must add two underscores (_) at both the start and end of source.
`__topic__`	Varchar	The topic of a log.

Limits

- Maximum number of Logstores from which you can query logs at the same time is 15.
- Maximum length for a field value of the varchar type is 2,048. Extra data will be truncated.
- By default, up to 100 lines of a log file are returned and pagination is not supported. If you want to return more lines, use LIMIT syntax.

Example

Calculate the hourly PV and UV, and the user request of the highest latency.

```
*|select date_trunc('hour',from_unixtime(__time__)) as time,
     count(1) as pv,
     approx_distinct(userid)  as uv,
     max_by(url,latency) as top_latency_url,
     max(latency,10) as top_10_latency
     group by  1
     order by time
```

28.1.4.3. Enable the indexing feature and configure indexes for a Logstore

This topic describes how to enable the indexing feature and configure indexes for a Logstore.

Context

Before you can query logs that are stored in a Logstore, you must enable the indexing feature and configure indexes for the Logstore. We recommend that you configure indexes for your Logstores based on your business requirements.

> **Note** After you enable the indexing feature, the indexes occupy extra storage space and transferring the indexes occupies extra bandwidth.

When you collect a log entry, Log Service adds the relevant information (such as the source and time fields) to the log entry as key-value pairs. These fields are reserved in Log Service. If you enable the indexing feature for a Logstore and configure indexes for fields in the Logstore, the indexing and analytics features are automatically enabled for these fields.

Reserved fields in Log Service

Field	Description
`__topic__`	The topic of a log entry. If you specify a topic for a log entry, Log Service adds the topic field to the log entry. The key of the field is __topic__ and the value of the field is the log topic. For more information, see `Specify a log topic`.
`__source__`	The source of a log entry. The source device that generates the log entry.
`__time__`	The time when a log entry is written to the Logstore by using an SDK.

> **Note** If the values of the `__topic__` and `__source__` fields are null, the keywords that you use to query the two fields must exactly match the field values.

Procedure

1. Log on to the Log Service console.
2. In the Projects section, click the target project.
3. Find the target Logstore, and choose > **Search & Analysis**.
4. On the page that appears, click **Enable** in the upper-right corner.

   > **Note** If you have created indexes for the Logstore, choose **Index Attributes** > **Modify** to modify the indexes.

5. Configure the indexes.

   > **Note** If you enable a full-text index and a field-specific index at the same time, the field-specific index takes precedence over the full-text index.

   Index types

Index type	Description
Full text index	An index is created in the text format for all fields. You can search for key-value pairs that are included in these fields. For fields of the LONG type, you must specify the key name of a field when you query a value of the field. For fields of the other types, you do not need to specify a key name in queries.

Index type	Description
Field-specific index	After you configure a field-specific index, you must specify the name of a key when you query logs. If you configure the field-specific index on a field, the field-specific index takes effect when you query logs. The full-text index does not take effect. Available data types that you can specify for fields include: ○ Query text data ○ JSON indexes ○ Numeric (LONG and DOUBLE)

- Configure a full-text index.

  After you configure a full-text index for a Logstore, the values of all fields in the Logstore are queried by default.

Parameter	Description	Example value
Full Text Index	If you turn on the switch, Log Service traverses the values of all fields in a log entry. If the value of one of the fields matches the keyword, the log entry is returned.	-
Case Sensitive	Specifies whether queries are case-sensitive. ■ If you turn off the switch, queries are not case-sensitive. For example, if you search for internalError, you can use either INTERNALERROR or internalerror as the keyword. ■ If you turn on the switch, queries are case-sensitive. For example, if you search for internalError, you can use only internalError as the keyword.	-
Include Chinese	Specifies whether to differentiate the Chinese content and English content. ■ If you turn on the switch, Log Service separates the Chinese content based on the Chinese semantics and English content based on the specified delimiters. ■ If you turn off the switch, the content of a log entry is separated by the specified delimiters.	-
Delimiter	The delimiters that you use to separate the content of a log entry into multiple keywords. For example, the content of a log entry is `a,b;c;D-F`. You can specify commas (,), semicolons (;), and hyphens (-) as delimiters to delimit the log content. Then you can use the five letters a, b, c, D, and F as keywords to match the log entry.	`, '";=()[]{}? @&<>/:\n\t`

- Configure a field-specific index.

  You can specify fields to be indexed. Field-specific indexes allow you to query log data based on the values of specific fields. This narrows down the query scope.

Parameter	Description	Example value
Key Name	The name of a log field. **Note** ■ If you want to configure an index for fields of the tag type, such as fields that include public IP addresses or UNIX timestamps, you must set the value of the **Key Name** parameter in the `__tag__:key` format, for example, `__tag__:__receive_time__`. ■ Indexes of the numeric types are unavailable for tag fields. You must select text in the **Type** field for all tag fields.	`_address_`
Type	The type of a field. Valid values: ■ **text**: The data type of the field is TEXT. ■ **long**: The data type of the field is LONG. You must specify a numeric range to query log data. ■ **double**: The data type of the field is DOUBLE. You must specify a numeric range to query log data. ■ **json**: The data type of the field is JSON. **Note** The **Case Sensitive**, **Include Chinese**, and **Delimiter** parameters are unavailable for fields of numeric types (LONG and DOUBLE).	-
Alias	The alias of a column. Aliases are applied only to SQL statistics. Original names are applied when you store and query log data in Log Service. For more information, see Field aliases.	`address`
Case Sensitive	Specifies whether queries are case-sensitive. ■ If you turn off the switch, queries are not case-sensitive. For example, if you search for internalError, you can use either INTERNALERROR or internalerror as the keyword. ■ If you turn on the switch, queries are case-sensitive. For example, if you search for internalError, you can use only internalError as the keyword.	-

Parameter	Description	Example value
Delimiter	The delimiters that you use to separate the content of a log entry into multiple keywords. For example, the content of a log entry is `a,b;c;D-F`. You can specify commas (,), semicolons (;), and hyphens (-) as delimiters to delimit the log content. Then you can use the five letters a, b, c, D, and F as keywords to match the log entry.	`, '";=()[]{}? @&<>/:\n\t`
Include Chinese	Specifies whether to differentiate the Chinese content and English content. ■ If you turn on the switch, Log Service separates the Chinese content based on the Chinese semantics and English content based on the specified delimiters. ■ If you turn off the switch, the content of a log entry is separated by the specified delimiters.	-
Enable Analytics	Specifies whether to enable the analytics feature. The switch is turned on by default. After you turn on the switch, you can use search and analytic statements to obtain statistical results.	-

6. Click **OK**.

> **Note**
> - The index configurations take effect within one minute.
> - After an index is enabled or modified, the updates on the index apply only to new data that is written to Log Service.

28.1.4.4. Query logs

This topic describes how to query logs in a Logstore. After you enable and configure the index of the Logstore, you can query and analyze logs in a Logstore in real time.

Prerequisites

- Logs are collected and stored in a Logstore.
- Indexes are enabled and configured. For more information, see Enable the index feature and configure indexes for a Logstore.

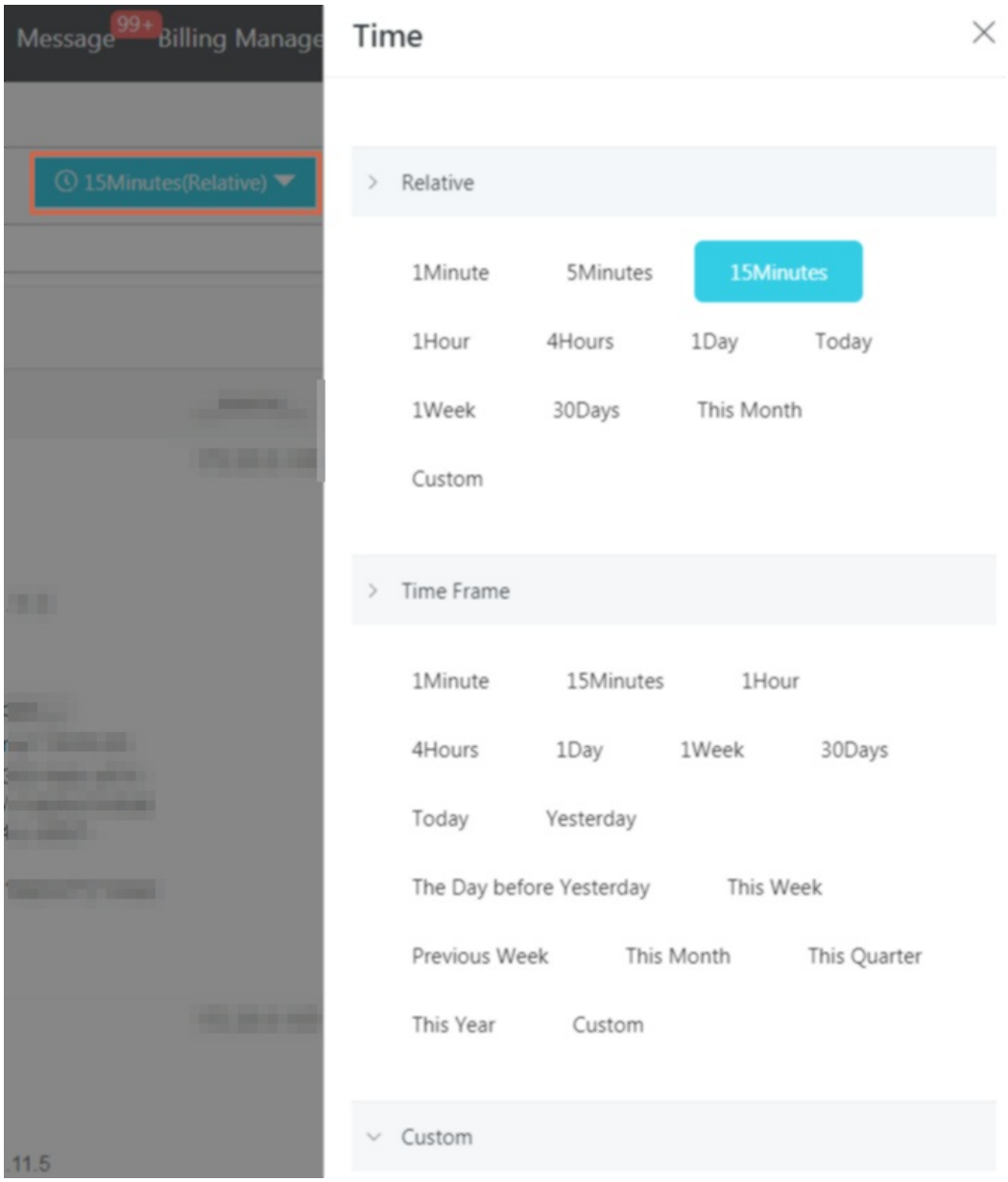
Procedure

1. Log on to the Log Service console.
2. Click the icon next to the name of the Logstore, and then select **Search & Analysis**.
3. Enter a query statement in the search box.

   A query statement consists of a search statement and an analytic statement. The syntax is `search statement|analytic statement`. For more information, see Search and analysis statements.
4. On the Search & Analysis page, select **15 Minutes** in the Relative section to set the time range for the query.

You can select a relative time, time frame, or a custom time range.

> **Note** The query results may contain logs that are generated 1 minute earlier or later than the specified time period.



5. Click **Query & Analyze** to view the query results.

   Log Service illustrates query results on log distribution histograms, Raw Logs tab, or statistical graphs.

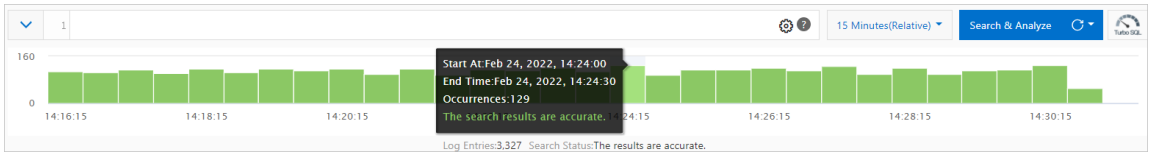
   > **Note** 100 results are returned by default. For information about how to retrieve more results, see LIMIT syntax.

   - Log distribution histogram

     The log distribution histogram shows the distribution of query results across different time ranges.

     - Move the pointer over a green block to view a time range and the number of logs obtained within the time range.

- Click a data block to view finer-grained log distribution. You can also view the query results on the **Raw Logs** tab.



- Raw Logs tab

  On the **Raw Logs** tab, view logs that match your search conditions.

  - Quick analysis: Use this feature to analyze the distribution of values for a specific field within a period of time. For more information, see Quick analysis.
  - Log download: Click the download icon in the upper-right corner of the tab, select a time range, and then click **OK**.
  - Column settings: Click **Column Settings** in the upper-right corner of the tab, select the required fields and click **Add** to add the fields. Then, the columns that correspond to the fields appear on the tab. The field names are also column names. The columns list the field values.

    > **Note** To view the log contents on the tab, select **Content**.

- Content column settings: If the content of a field exceeds 3,000 characters, extra characters will be hidden. In this case, the message "The character string is too long and has been truncated" will be displayed before the Key field. Click **Display Content Column**. In the dialog box that appears, set the **Key-Value Arrangement** and **Truncate Character String** parameters.

  > **Note** If the content limit is set to 10,000 characters, no delimiter will be specified for extra characters.

Parameter		Description
Key-Value Pair Arrangement		You can set this parameter to **New Line** or **Full Line**.
Truncate Character String	**Key**	If a field value contains more than 3,000 characters, the field value is truncated. However, this parameter remains unspecified if no field value exceeds 3,000 characters. The value of this parameter is the key of the truncated value.
	Status	This parameter determines whether to enable the value truncation feature. By default, the feature is enabled. ■ **Enable**: If the value in a key-value pair exceeds the specified **Truncate Step**, extra characters will be truncated. You can click the Show button at the end of the value to show the truncated characters. The increment per click is the specified truncate step. ■ **Disable**: If the value in the key-value pair exceeds the specified **Truncate Step**, extra characters will not be truncated.
	Truncate Step	This parameter specifies the maximum number of characters that a field value shows by default. The parameter also specifies the number of extra characters that you displayed each time you click the Show button. Valid values: 500 to 10000. Default value: 3000.

- Graph

  If you enable the Analytics feature on the Search & Analysis page and use search and analytic statements to query logs, you can view the analytical results on the **Graph** tab.

  - Graphs of multiple types are provided in Log Service, including tables, line charts, and bar charts. You can select a graph to show the required analytical results. For more information, see Graphs.
  - Log Service allows you to create dashboards for real-time data analysis. For more information, see Create and delete a dashboard. Click **Add to New Dashboard** to save a common chart as query statements to a dashboard.
  - Drill-down analysis allows you to move to deeper data layers, which reveals more detailed information. You can set the drill-down parameters and add the chart to the dashboard. Then, you can click the values in the chart to view the analysis results from more dimensions. For more information, see Drill-down analysis.

  You can also click **Save Search** or **Save as Alarm** on the Search & Analysis page to use the saved search and alarm features. For more information, see Save a query statement as a search and Configure an alert.

28.1.4.5. Export logs

You can export logs on the current page to a CSV file and save the file to your localhost.

Procedure

1. Log on to the Log Service console.
2. Click a project name.
3. Click the icon next to the name of the Logstore, and then select **Search & Analysis**.
4. Click the icon next to the **Raw Logs** tab.
5. In the **Download Log** dialog box, select **Download Log in Current Page**.
6. Click **OK** to export logs of the current page to a .CSV file and save the file to the localhost.

28.1.4.6. Index data type

28.1.4.6.1. Overview

Log Service allows you to use full-text indexes or field-specific indexes to query collected logs. If you set a full-text index for a log, the value is the entire log. If you set a field-specific index for a log, you can specify a data type for each key.

Date types

The following table lists the supported data types.

Query type	Data type (index)	Description	Example
Basic query	Text	The text type. You can use keywords and fuzzy matches to query logs.	`uri:"login*" method:"post"`
	Long	The numeric type. You can specify numeric ranges to query logs.	`status>200 and status in [200, 500]`
	Double	The floating-point type.	`price>28.95 and t in [20.0, 37]`
Combined query	JSON	Indicates that the index is a JSON field that supports nested queries. By default, the data type of the field is text. You can set indexes of the Text, Long, and Double types for the b elements at layer a in the a.b path format. The fields adopt the configured types.	`level0.key>29.95 level0.key2:"action"`
	Text	Indicates that the full contents of the log are queried as text.	`error and "login fail"`

Query examples

The following table lists the keys included in the sample log.

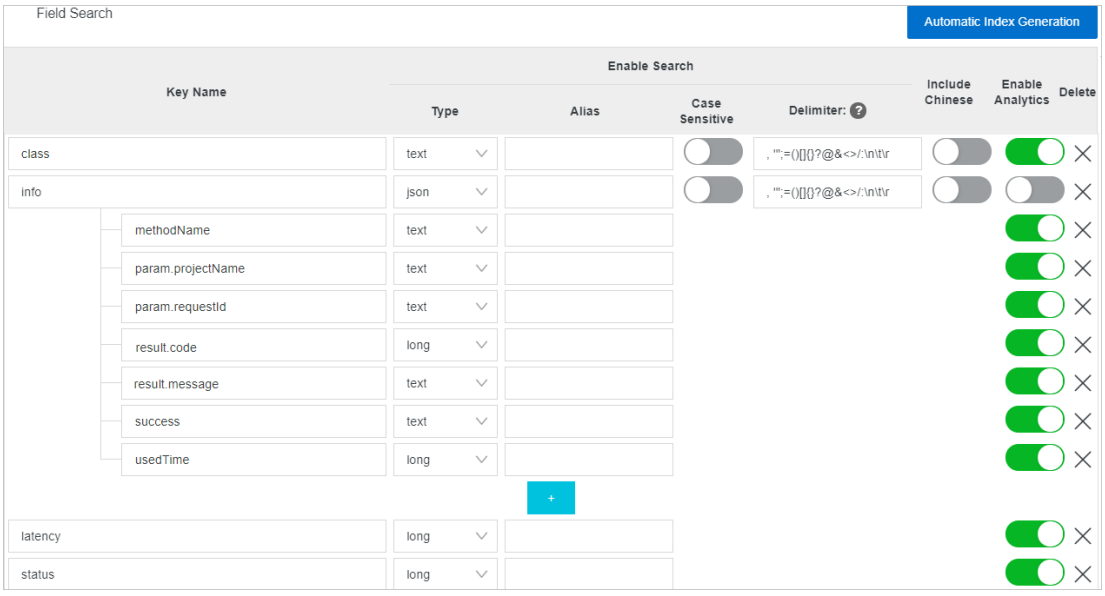
No.	Key	Type
0	time	N/A
1	class	text

No.	Key	Type
2	status	long
3	latency	double
4	message	json

```
0. time:2018-01-01 12:00:00
  1. class:central-log
  2. status:200
  3. latency:68.75
  4. message:
  {
      "methodName": "getProjectInfo",
      "success": true,
      "remoteAddress": "1.1.1.1:11111",
      "usedTime": 48,
      "param": {
              "projectName": "ali-log-test-project",
              "requestId": "d3f0c96a-51b0-4166-a850-f4175dde7323"
      },
      "result": {
          "message": "successful",
          "code": "200",
          "data": {
              "clusterRegion": "ap-southeast-1",
              "ProjectName": "ali-log-test-project",
              "CreateTime": "2017-06-08 20:22:41"
          },
          "success": true
      }
  }
```

You can set an index as follows.

Set an index



In the preceding figure,

- ① specifies that Log Service can query data of the string and Boolean types in JSON fields.
- ② specifies that Log Service can query data of the long type.
- ③ enables SQL analysis for specified fields.

Example

1. Query data of the string and Boolean types
   - You do not need to configure JSON fields.
   - JSON maps and arrays are automatically expanded and can contain nested fields. Separate multiple levels with periods (`.`). .

   ```
   class : cental*
   message.traceInfo.requestId : 92.137_1518139699935_5599
   message.param.projectName : ali-log-test-project
   message.success : true
   ```

2. Query data of the double and long types

   Each JSON field must be specified separately and cannot be contained in a JSON array.

   ```
   latency>40
   message.usedTime > 40
   ```

3. Combined query

   ```
   class : cental* and message.usedTime > 40 not message.param.projectName:ali-log-test-project
   ```

28.1.4.6.2. Query text data

This topic describes how to query text data.

Similar to search engines, Log Service queries text data based on terms. Therefore, you must set the Delimiter, Case Sensitive fields.

Configurations

- Case sensitivity

  You can specify whether log queries are case-sensitive. For example, you want to query logs by using a search term named `internalError` .

  - *false* specifies a case-insensitive query. Both `INTERNALERROR` and `internalerror` can be the keywords.
  - *true* specifies a case-sensitive query. Only the `internalError` can be the keyword.

- Delimiter

  You can use delimiters to split a search term into multiple keywords.

  For example, you want to query logs by using the following search term.

  ```
  /url/pic/abc.gif
  ```

  - If no delimiter is set, the entire `/url/pic/abc.gif` string is treated as a keyword. You must use the entire string as a keyword or a fuzzy string named `/url/pic/*` to query logs.
  - If the delimiter is set to `/` , the search term is split into three words: `url` , `pic` , and `abc.gif` . You can use one of these words or a fuzzy word to query logs. For example, `url` , `abc.gif` , or `pi*` . You can also use the `/url/pic/abc.gif` string as a search term to query logs. However, the search term is split into three keywords named `url, pic, and abc.gif` .
  - If the delimiter is set to `/.` , the search term is split into four keywords named `url` , `pic` , `abc` , and `gif` .

> **Note** You can extend query ranges by setting appropriate delimiters.

- Full-text index

  By default, full-text indexes treat each log except for the time field as text data. You do not need to specify any keys for a full-text index. For example, the following log includes the time field, status field, level field, and message field.

  ```
  [20180102 12:00:00] 200,error,some thing is error in this field
  ```

  - `time:2018-01-02 12:00:00`
  - `level:"error"`
  - `status:200`
  - `message:”some thing is error in this field”`

  > **Note**
  > - Prefixes are not required for full-text indexes. If you set the search term to error, the level and message fields that include error match the search term.
  > - You must set delimiters for full-text indexes. For example, if you set a space ( ) as a delimiter, the `status:200` string is a search term. If you set a colon (`:`) as a delimiter, the search term is split into two keywords named `status` and `200` .
  > - Numbers are treated as text data. For example, you can use 200 to query logs. Values in the time field are not treated as text data.
  > - You can query logs by using keys such as `status` .

28.1.4.6.3. Numeric type

When you configure indexes, you can set the data type of a key to number. To query logs, you can specify a numeric range for the key.

Configurations

Supported types: `long` (long integers) and `double` (decimals). After you set the data type of a key to number, you must specify a numeric range for the key to query logs.

Query examples

To specify a numeric range from 1000 to 2000 (excluding 1000) for a key of the long type, you can use the following methods:

- Query syntax for numbers. For example:

  ```
  longKey > 1000 and longKey <= 2000
  ```

- Query grammar for numeric ranges. For example:

  ```
  longKey in (1000 2000]
  ```

For more information about query syntax, see Query syntax.

28.1.4.6.4. JSON indexes

Log Service can query and analyze logs in the JSON format. You can set the data type of indexes to JSON.

JSON texts include data of multiple types, including string, Boolean, number, array, and map. JSON-formatted data is self-parsed and flexible. You can use JSON-formatted data in various scenarios. In most cases, variable log fields are recorded in the JSON format. For example, HTTP request and response parameters are recorded in a log in the JSON format.

Log Service allows you to set the data type of index fields to JSON so that you can query and analyze logs in the JSON format.

Configurations

- Log Service can parse JSON-formatted fields and generate indexes for all the fields of the text and Boolean types.

```
json_string.key_map.key_text : test_value
json_string.key_map.key_bool : true
```

- To query fields of the double or long type that is not in a JSON array, you can specify a JSON path.

```
Set the data type of the key_map.key_long field to long.
Search condition: json_string.key_map.key_long > 50
```

- To query fields of the text, double, or long type that is not in a JSON array, you can enable the Analytics feature and use SQL statements to analyze these fields.

```
json_string.key_map.key_long > 10 | select count(*) as c ,
    "json_string.key_map.key_text" group by
    "json_string.key_map.key_text"
```

> **Note**
>
> - JSON objects and JSON arrays are not supported.
> - Fields cannot be contained in JSON arrays.
> - Fields of the Boolean type can be converted into the text type.
> - To query and analyze logs, JSON-formatted fields must be enclosed with double quotation marks (" ").

- Log Service cannot parse invalid JSON-formatted data.

  Log Service does not stop parsing logs until it detects an invalid field.

  In the following example, data after the key_3 field is truncated and lost in the following text. Log Service can parse the json_string.key_map.key_2 field and the contents before this field.

```
"json_string":
{
     "key_1" :  "value_1",
     "key_map" :
      {
             "key_2" : "value_2",
             "key_3" : "valu
```

Query syntax

To query a specific key, you must add the JSON parent path to the query statement as the prefix of the key. The query syntax for the fields of the text and numeric types is the same for both JSON-formatted data and other data. For more information, see Query syntax.

Query example

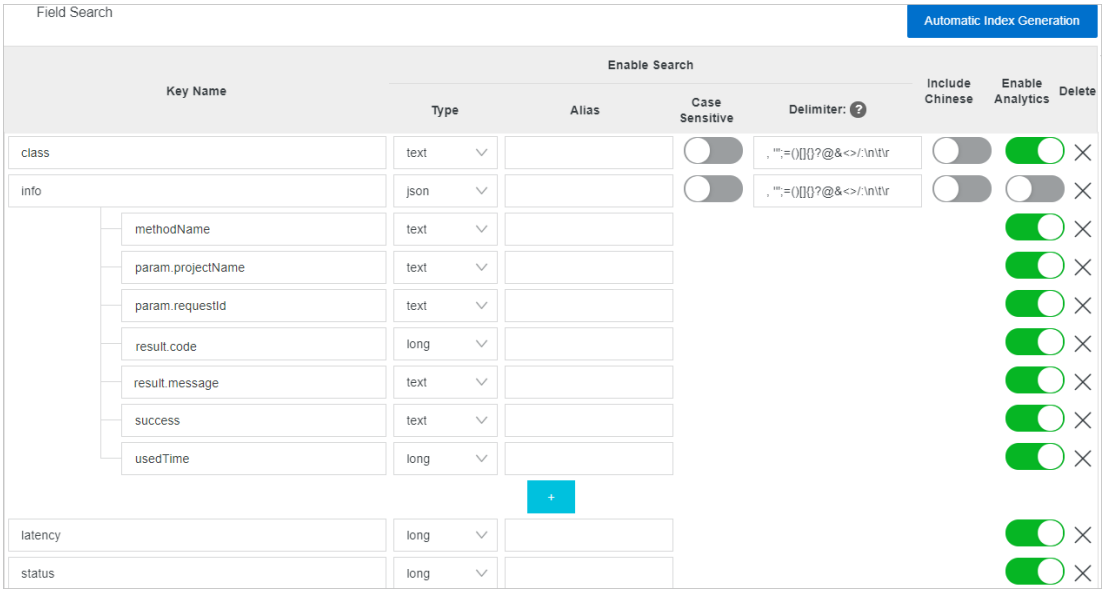
The following table lists the keys included in the sample log. The data type of the `message` key is JSON.

Number	Key	Type
0	time	N/A
1	class	text
2	status	long
3	latency	double
4	message	json

```
0. time:2018-01-01 12:00:00
  1. class:central-log
  2. status:200
  3. latency:68.75
  4. message:
  {
      "methodName": "getProjectInfo",
      "success": true,
      "remoteAddress": "1.1.1.1:11111",
      "usedTime": 48,
      "param": {
              "projectName": "ali-log-test-project",
              "requestId": "d3f0c96a-51b0-4166-a850-f4175dde7323"
      },
      "result": {
          "message": "successful",
          "code": "200",
          "data": {
              "clusterRegion": "ap-southeast-1",
              "ProjectName": "ali-log-test-project",
              "CreateTime": "2017-06-08 20:22:41"
          },
          "success": true
      }
  }
```

You can set indexes for the log as follows:

Set an index



In the preceding figure:

- ① specifies that Log Service can query data of the string and Boolean types in JSON fields.
- ② specifies that Log Service can query data of the long type.
- ③ enables SQL analysis for specified fields.

Examples

1. Query data of the string and Boolean types

   > **Note**
   > - You do not need to configure JSON fields.
   > - JSON maps and arrays are automatically expanded and can include hierarchical levels. Separate multiple levels with periods (`.`).

   ```
   message.traceInfo.requestId : 92.137_1518139699935_5599
   message.param.projectName : ali-log-test-project
   message.success : true
   message.result.data.ProjectStatus : Normal
   ```

2. Query fields of the double and long types

   > **Note** Each JSON field must be configured and cannot be contained in an array.

   ```
   message.usedTime > 40
   ```

3. Use SQL statements to analyze fields

   > **Note**
   > - Each JSON field must be configured and cannot be contained in an array.
   > - Each field to be queried must be enclosed with double quotation marks (" ") or be configured with an alias.

   ```
   * | select avg("message.usedTime") as avg_time ,
   "message.methodName"  group by "message.methodName"
   ```

28.1.4.7. Query syntax and functions

28.1.4.7.1. Search syntax

This topic describes the search syntax that is used in Log Service.

Search types

After you enable and configure the index feature of a Logstore, you can enter a search statement on the search and analysis page to query logs.

A query statement consists of two sub-statements in sequence: a search statement and an analytic statement. A search statement specifies one or more search conditions and returns the log entries that match the search conditions. You can execute a search statement to perform a full-text search or field-specific search.

- Full-text search

  During full-text search, a log entry is considered a key-value pair. The value in the key-value pair indicates the content of the log entry. A full-text search statement returns the log entries that include or exclude the specified keywords.

  Full-text search is divided into basic full-text search, phrase search, and wildcard-based search.

  - Basic full-text search: You can specify keywords and operators in search conditions of a search statement. You can then execute the search statement to query the log entries that match the search conditions.

    For example, the `a and b` statement returns the log entries that include the `a` and `b` keywords.

  - Phrase search: A phrase is a string that is enclosed in double quotation marks (""). Substrings in a phrase are separated by space characters. Each substring is a keyword.

    For example, the `"http error"` statement returns the log entries that contain the http and error keywords. This statement is equivalent to `http and error`.

  - Wildcard-based search: You can use an asterisk (`*`) or a question mark (`?`) as a wildcard character in a keyword. Each keyword that includes wildcards can contain 1 to 64 characters in length and cannot start with a wildcard character. If a search condition contains a keyword that includes a wildcard character, Log Service returns a maximum of 100 log entries and each log entry contains a word that matches the keyword pattern.

    For example, if you execute the `addr?` statement, Log Service returns a maximum of 100 log entries and each log entry contains a word that is prefixed with `addr`.

    When you use wildcard-based search, note the following information:

    - A keyword cannot start with an asterisk (`*`) or a question mark (`?`).
    - The more accurate the keyword is, the more accurate the search results will be.
    - Wildcard-based search is not supported for a keyword that contains more than 64 characters in length.
    - A search statement returns a maximum of 100 log entries that match the search conditions.

- Field-specific search

  After you configure the field index, you can search log entries based on the keys and values of the fields in the field index. For a field of the DOUBLE or LONG type, you can specify a value range for search. For example, the `Latency>5000 and Method:Get* and not Status:200` statement returns the log entries that meet the following conditions: The value of the `Latency` field is greater than 5000, the value of the `Method` field is prefixed with `Get`, and the value of the `Status` field is not 200.

  You can perform a basic query or combined query, depending on the data types of the fields in the field index. For more information, see Overview.

Additional considerations

- If you execute a search statement to perform both full-text search and field-specific search and you set different delimiters for the two search types, the delimiter that is set for field-specific search is used.
- You must set the data type of a field to DOUBLE or LONG before you specify a value range to search the field. If the data type of a field is not DOUBLE or LONG or the value range syntax is incorrect, the field-specific search condition is considered a full-text search condition. In this case, unexpected search results may be returned.
- If you change the data type of a field from TEXT to DOUBLE or LONG, only the equal-to operator (`=`) can be used to search for the log entries that are collected before the change.

Operators

The following table lists the operators that are supported by search statements.

Operator	Description
and	A binary operator. The syntax is `query1 and query2`. It indicates the intersection of the search results of `query1` and `query2`. The default operator between keywords is `and`.
or	A binary operator. The syntax is `query1 or query2`. It indicates the union of the search results of `query1` and `query2`.
not	A binary operator. The syntax is `query1 not query2`. It indicates that the log entries that match `query1` but do not match `query2` are returned. The syntax is equivalent to `query1-query2`. You can also use the `not query1` syntax. It indicates that the log entries that do not match `query1` are returned.
(,)	The operator that merges one or more sub-conditions into one search condition. The search based on a sub-condition that is enclosed in parentheses () is performed first.
:	The operator that is used to specify a pattern of key-value pairs. The syntax is `term1:term2`. If the key or value contains reserved characters such as spaces and colons (:), use double quotation marks (`""`) to enclose the entire key or value.
"	The operator that converts another operator into a common character. All terms enclosed in double quotation marks ("") are considered keywords rather than operators. In a field-specific search statement, you can enclose the entire key or value in double quotation marks.
\	The operator that escapes a double quotation mark. The escaped double quotation mark is considered a symbol instead of an operator. Example: `"\""`.
|	The pipeline operator that is used to chain a search statement and an analytic statement. The analytic statement that follows the pipeline operator is executed based on the result of the search statement that the pipeline operator follows. Example: `query1 | select count(1)`.
count	The count operator that is used to summarize the number of log entries.
*	The wildcard character that is used to replace zero or more characters. For example, the `que*` statement returns the log entries with a word that is prefixed with `que`. **Note** A wildcard-based search statement returns a maximum of 100 log entries that match the search condition.

Operator	Description
?	The wildcard character that replaces a single character. The `qu?ry` statement returns the log entries with a word that is prefixed with `qu`, is suffixed with `ry`, and contains a character in between.
`__topic__`	The operator that specifies zero or more topics from which to query log entries. Example: `__topic__:mytopicname`.
`__tag__`	The operator that specifies a tag value of a tag key to query. Example: `__tag__:tagkey:tagvalue`.
source	The operator that specifies the IP address of a log source whose log entries you want to query. Example: `source:127.0.0.1`.
>	The greater-than operator. You can use this operator to query the log entries whose value of a field is greater than a specified number. Example: `latency > 100`.
>=	The greater-than-or-equal-to operator. You can use this operator to query the log entries whose value of a field is greater than or equal to a specified number. Example: `latency >= 100`.
<	The less-than operator. You can use this operator to query the log entries whose value of a field is less than a specified number. Example: `latency < 100`.
<=	The less-than-or-equal-to operator. You can use this operator to query the log entries whose value of a field is less than or equal to a specified number. Example: `latency <= 100`.
=	The equal-to operator. You can use this operator to query the log entries whose value of a field is equal to a specified number. Example: `latency = 100`.
in	The operator that is used to query the log entries whose value of a field falls in a specified range. Brackets [] indicate closed intervals and parentheses () indicate open intervals. The beginning number and ending number of the range are enclosed in brackets or parentheses and separated by one or more space characters. The in operator must be in lowercase. Example: `latency in [100 200]` or `latency in (100 200]`.

Note

- All operators except the in operator are case-insensitive.
- You can use the following operators, which are sorted in descending order of precedence: `:`, `"`, `()`, `and`, `not`, and `or`.
- Log Service uses the following operators: `sort`, `asc`, `desc`, `group by`, `avg`, `sum`, `min`, `max`, and `limit`. If you need to use these operators as keywords, enclose them in double quotation marks ("").

Search statement examples

Expected search result	Search statement
Log entries that contain a and b	`a and b` or `a b`
Log entries that contain a or b	`a or b`

Expected search result	Search statement
Log entries that contain a but do not contain b	`a not b`
Log entries that do not contain a	`not a`
Log entries that contain a and b but do not contain c	`a and b not c`
Log entries that contain a or b and contain c	`(a or b ) and c`
Log entries that contain a or b but do not contain c	`(a or b ) not c`
Log entries that contain a and b and may contain c	`a and b or c`
Log entries whose FILE field contains apsara	`FILE:apsara`
Log entries whose FILE field contains apsara and shennong	`FILE:"apsara shennong"` , `FILE:apsara FILE:shennong` , or `FILE:apsara and FILE:shennong`
Log entries that contain the following keyword: and	`and`
Log entries whose FILE field contains apsara or shennong	`FILE:apsara or FILE:shennong`
Log entries whose file info field contains apsara	`"file info":apsara`
Log entries that contain double quotation mark (")	`\"`
Log entries with words that are prefixed with shen	`shen*`
Log entries whose FILE field is prefixed with shen	`FILE:shen*`
Log entries whose value of the FILE field is shen*	`FILE: "shen*"`
Log entries with words that are prefixed shen, are suffixed with ong, and contain a single character in between	`shen?ong`
Log entries with words that are prefixed with shen and words that are prefixed with aps	`shen* and aps*`
Log entries of topic1 and topic2	`__topic__:topic1 or __topic__ : topic2`
Log entries with a tag whose key is tagkey1 and value is tagvalue2	`__tag__ : tagkey1 : tagvalue2`
Log entries whose value of the latency field is greater than or equal to 100 and less than 200	`latency>= 100 and latency < 200` or `latency in [100 200)`
Log entries whose value of the latency field is greater than 100	`latency > 100`
Log entries that do not contain spider and whose http_referer field does not contain opx	`not spider not bot not http_referer:opx`
Log entries whose cdnIP field is not empty	`not cdnIP:""`
Log entries that do not contain the cdnIP field	`not cdnIP:*`

Expected search result	Search statement
Log entries that contain the cdnIP field	`cdnIP:*`
Log entries that contain a specified URL	`* | select * where url = 'www.xxxxx.com'`

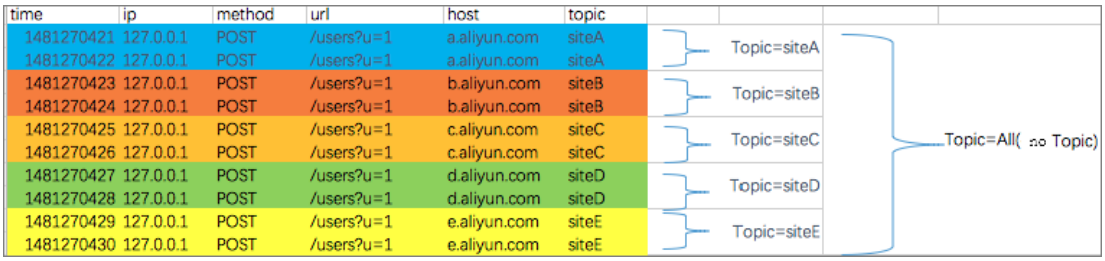
Topic-specific search

Each Logstore is divided into one or more topics. You can divide a Logstore into multiple topics if you need level-2 categories of log entries. When you query logs, you can specify topics to increase efficiency.

In a search statement, you can specify one or more topics to query. If no topic is specified, log entries are queried from all topics.

For example, you can classify log entries into multiple topics based on domain names.

Log topics



Syntax of topic-specific search:

- In a search statement, you can specify one or more topics to query. If no topic is specified, log entries are queried from all topics.
- The topic-specific search syntax is `__topic__:topicName`. You can also specify a topic in a URL.
- You can query log entries from multiple topics. For example, the `__topic__:topic1 or __topic__:topic2` statement returns the log entries in topic1 and topic2.

28.1.4.7.2. LiveTail

This topic describes how to use LiveTail to monitor and analyze log data. LiveTail is an interactive feature provided in the Log Service console to monitor and extract key log data in real time.

Prerequisites

- LiveTail is available only after logs are collected.
- LiveTail can only monitor and extract log data collected by Logtail.

Context

In online O&M scenarios, you often need to monitor collected log data in real time and extract key information from the latest log data to locate error causes. In traditional O&M, you must run the `tail -f` command on servers to monitor log files in real time. To easily obtain the required real-time log information, you can include the `grep` or `grep -v` command to filter log entries by keyword. To simplify online O&M, Log Service provides LiveTail in the console to monitor and analyze online log data in real time.

Benefits

- Monitors real-time log information and filters log data by keyword.
- Logs collected based on the log collection configurations are identified by index.
- Log fields are delimited. This allows you to query contextual logs that contain delimiters.
- Log files can be tracked based on a single log entry and monitored in real time without the need to connect to

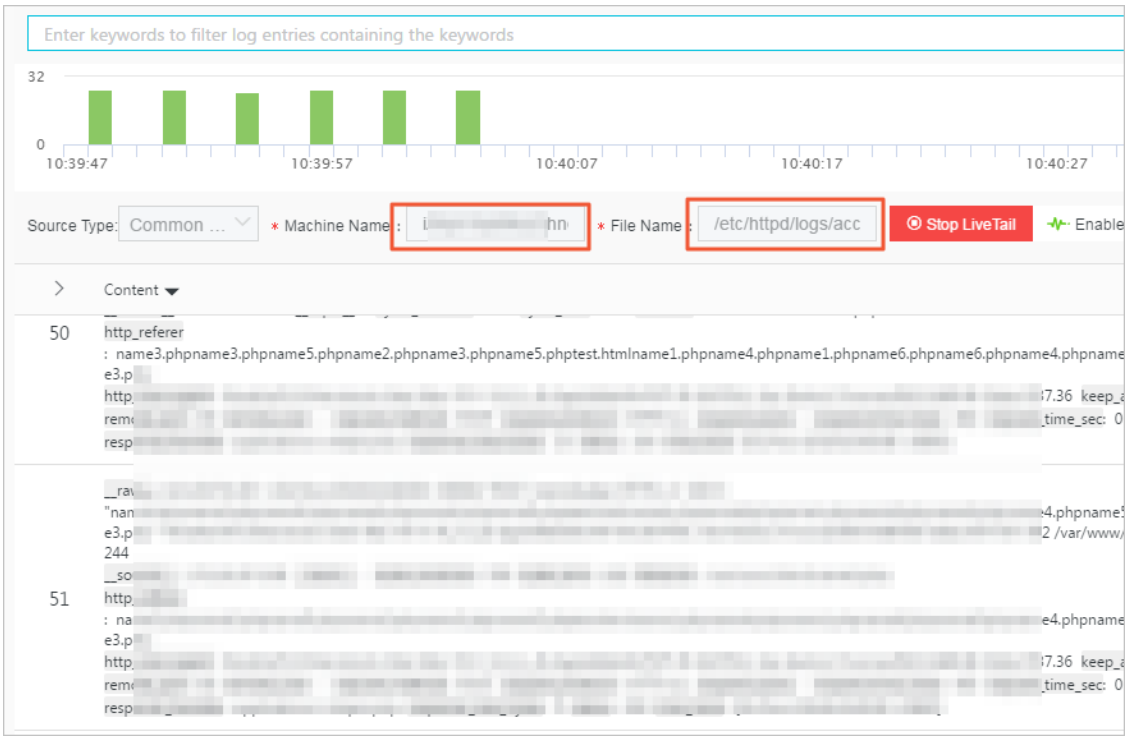
online servers.

Use LiveTail to monitor logs in a Logstore in real time

1. Log on to the Log Service console.
2. In the Projects section, click the target project.
3. Click the icon next to the name of the Logstore, and then select **Search & Analysis**.
4. (Optional)Start LiveTail.
   i. On the **Raw Logs** tab, click the icon next to the sequence number of the specified raw log entry, and then select **LiveTail**.

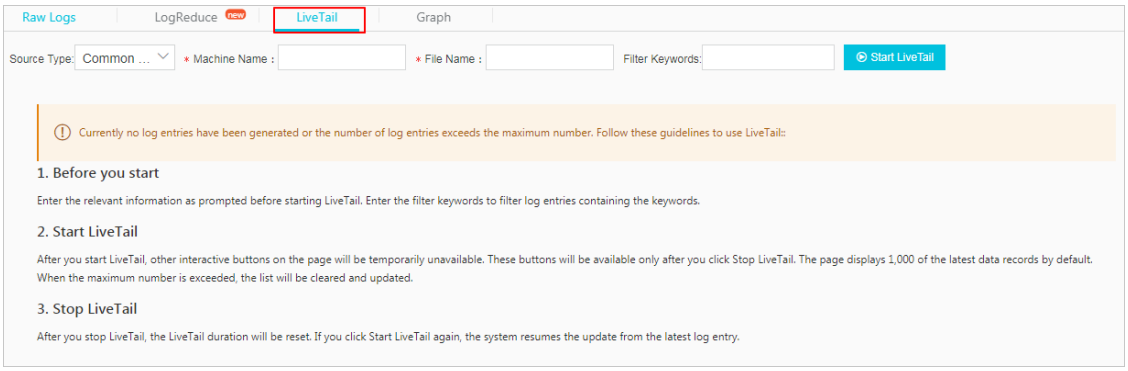
      The system starts LiveTail and starts timing. The **Source Type**, **Machine Name**, and **File Name** fields are automatically filled in based on the specified raw log entry.

      After LiveTail is started, log data collected by Logtail is displayed in order on the page in real time. The latest log data is displayed at the bottom of the page by default. You can view the latest log data without the need to drag the scroll bar. Up to 1,000 log entries can be displayed on the page. If more than 1,000 log entries are collected, the page is automatically refreshed to show the latest 1,000 log entries.

   ii. (Optional)You can also enter keywords in the search box to display log entries that contain the keywords in the monitoring list. By filtering log entries that contain the keyword, you can monitor specific log entries in real time.
5. Customize LiveTail.

i. On the Search & Analysis page, click the **LiveTail** tab.



ii. Configure LiveTail.

Parameter	Required	Description
Source Type	Yes	The source of log entries. Valid values: ■ Physical servers ■ Kubernetes containers ■ Docker
Machine Name	Yes	The name of the server from which log entries are collected.
File Name	Yes	The full path and name of the log file.
Filter Keywords	No	A keyword. After you set a keyword, only log entries that contain the keyword are displayed in the log monitoring list.

iii. Click **Start LiveTail**.

After LiveTail is started, log data collected by Logtail is displayed in order on the page in real time. The latest log data is displayed at the bottom of the page by default. You can view the latest log data without the need to drag the scroll bar. Up to 1,000 log entries can be displayed on the page. When more than 1,000 log entries are collected, the page is refreshed to show the latest 1,000 log entries.

6. To analyze logs during real-time log monitoring, click **Stop LiveTail**.

After you stop LiveTail, the LiveTail timing and the real-time log data update also stop.

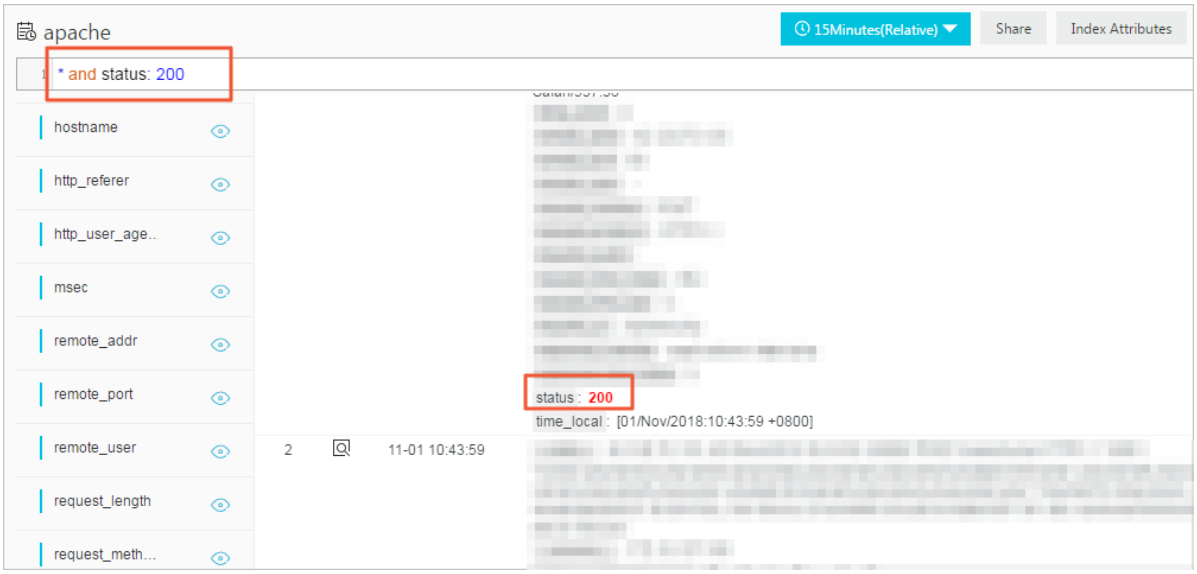
Log Service provides multiple methods to analyze exceptions that are found during log monitoring. For more information, see Use LiveTail to analyze logs.

Use LiveTail to analyze logs

After you stop LiveTail, real-time log updates in the log monitoring list also stop. You can analyze and fix errors that are found during log monitoring.

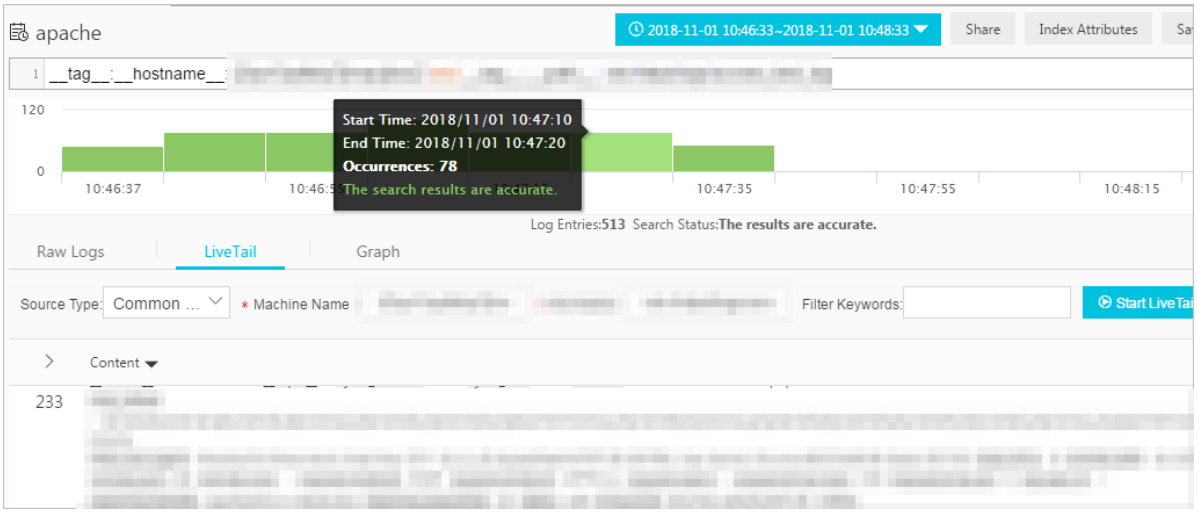
- View log entries that contain specified fields

  Log fields are delimited. You can click the value of the specified exception field on the **LiveTail** tab. Then, you are automatically forwarded to the **Raw logs** tab and the value of the exception field is used as a keyword to filter all log entries that contain the field and the keyword. You can also analyze log entries that contain the keyword based on contextual queries and statistical charts.



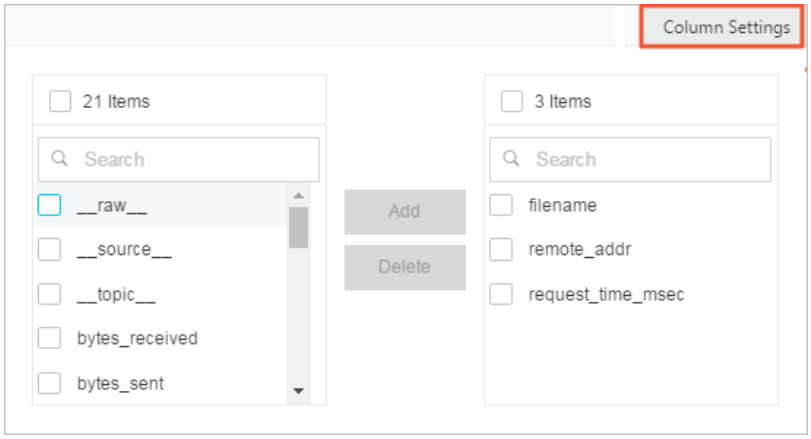
- Narrow down the time range of a log query based on the log distribution histogram

  After LiveTail is started, the log distribution histogram is updated at the same time. If you find a distribution exception (for example, a big increase in the number of log entries) in a time range, you can click the corresponding green rectangle to narrow down the time range of the log query. The timeline of the raw logs is associated with the timeline that you click on the LiveTail tab. You can view all relevant raw logs and log distribution details during this time range.



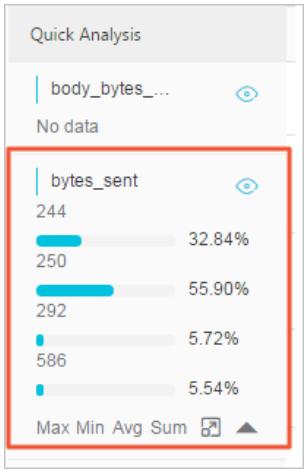
- Highlight key information based on column settings

  On the **LiveTail** tab, click **Column Settings** in the upper-right corner of the log monitoring list. Then, you can specify a field as a separate column to highlight relevant data.



- Analyze log data

  On the **LiveTail** tab, click the arrow in the upper-left corner of the log monitoring list to show the Quick Analysis pane. The time range for quick analysis starts from the time when LiveTail is started and ends at the time when LiveTail is stopped. The quick analysis provided on the LiveTail tab is the same as that provided on the Raw Logs tab. For more information, see Quick analysis.



28.1.4.7.3. LogReduce

This topic describes how to use LogReduce to group log data in Log Service. The LogReduce feature groups similar log entries by detecting same log patterns during text log collection.

Context

The LogReduce feature allows you to group text logs of multiple formats. You can locate errors, detect anomalies, roll back versions, and perform other O&M operations in DevOps scenarios. You can also detect network intrusions to ensure data security. In addition, you can save the log grouping result to a dashboard as an analysis chart, and then view the grouped data in real time.

Benefits

- Various formats of logs such as Log4j logs, JSON-formatted logs, and syslog logs can be grouped.
- Hundreds of millions of log entries can be grouped in seconds.
- Log entries can be grouped in any pattern.
- Raw log entries can be retrieved based on the signature of log entries grouped in a pattern.
- The number of log entries grouped in a log pattern in different time ranges can be compared.
- The precision of log grouping can be adjusted based on your needs.

Billing method

After the LogReduce feature is enabled, the size of indexes increases by 10% of the raw log size. For example, if the size of raw log data is 100 GB per day, the size of log indexes increases by 10 GB.

Raw log size	Index percentage	Size of indexes generated by LogReduce	Index size
100 GB	20% (20 GB)	100 × 10%	30 GB
100 GB	40% (40 GB)	100 × 10%	50 GB
100 GB	100% (100 GB)	100 × 10%	110 GB

Enable LogReduce of a Logstore

The LogReduce feature is disabled by default.

1. Log on to the Log Service console.
2. Click the target project in the Projects section.
3. Click the icon next to the name of the Logstore, and then select **Search & Analysis**.
4. Configure an index.
   - If you have enabled the index feature and configured indexes for the Logstore, choose **Index Attributes > Modify**.
   - If you have not enabled the index feature, click **Enable**.
5. Set the index attributes and turn on the **LogReduce** switch. Click **OK**.

   After LogReduce is enabled, Log Service groups log data that has been collected. Then, you can perform the following operations:
   - View log grouping results and raw logs
   - Adjust the precision of log grouping
   - Compare the number of log entries grouped in different time ranges

View log grouping results and raw logs

1. On the Search & Analysis page, enter a search and analytic statement in the search box, and then click **Search & Analytics**.

   You can use keywords to filter grouped log entries.

   > **Note** SQL statements are not supported. This means analysis results cannot be grouped.

2. Click the **LogReduce** tab to view the log grouping result.

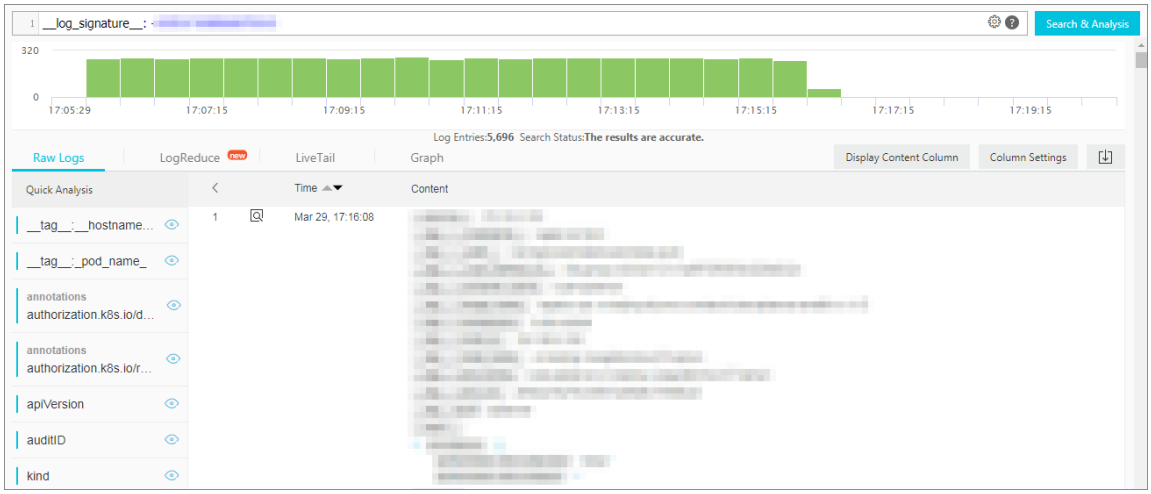
   The filtered log grouping result is displayed on the **LogReduce** tab.

Item	Description
Number	The sequence number of a log group.
Count	The number of log entries in a log group.
Pattern	The log pattern. Each log group has one or more sub-patterns.

   - Move the pointer over a value in the **Count** column to view the sub-patterns of the corresponding log group and the percentage of each sub-pattern in the log group. You can also click the plus sign (+) before

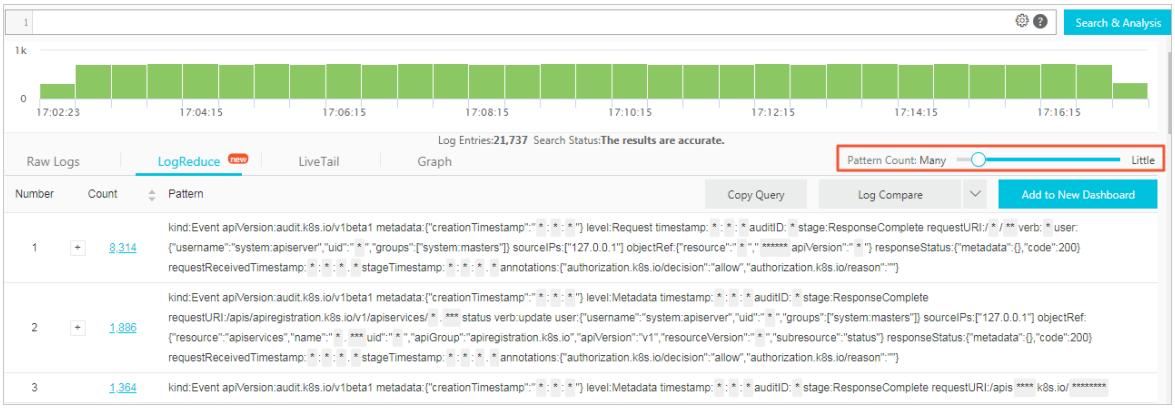
the count value to expand the sub-pattern list.

- Click a count value to view the raw log entries of the corresponding log group.



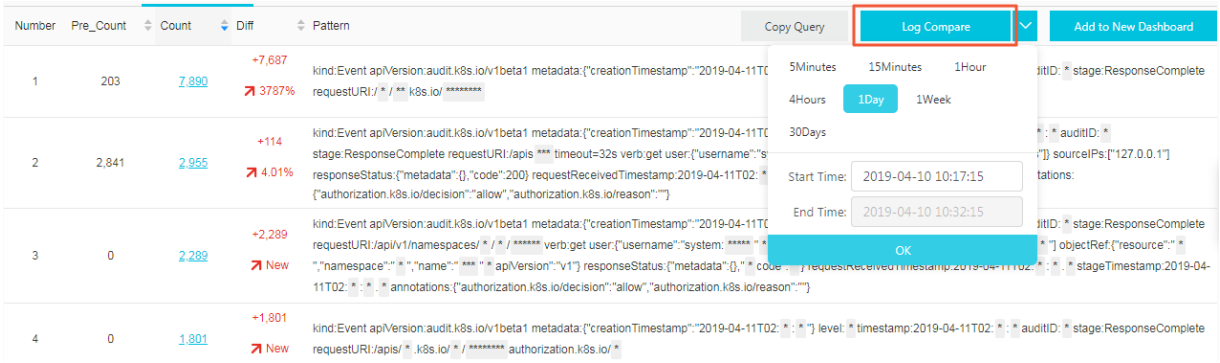
Adjust the precision of log grouping

1. On the Search & Analysis page, click the **LogReduce** tab.
2. In the upper-right corner of the tab, drag the **Pattern Count** slider to adjust the precision of log grouping.
   - If you drag the slider towards **Many**, you can obtain a more precise log grouping result with more detailed patterns.
   - If you drag the slider towards **Little**, you can obtain a less precise log grouping result with less detailed patterns.



Compare the number of log entries grouped in different time ranges

Click **Log Compare** on the **LogReduce** tab, select a time range, and then click **OK**.



Item	Description
Number	The sequence number of a log group.
Pre_Count	The number of log entries grouped by the current pattern in the previous time range.
Count	The number of log entries grouped by the current pattern in the current time range.
Diff	The difference between the Pre_Count value and Count value.
Pattern	The pattern of a log group.

SQL statement examples:

- Obtain a log grouping result.
  - SQL statement:

    ```
    * | select a.pattern, a.count,a.signature, a.origin_signatures from (select log_reduce(3) as a from log) limit 1000
    ```

    > **Note** When you view the log grouping result in the Log Service console, you can click **Copy Query** to obtain the relevant SQL statement.

  - Input parameter: log_reduce (precision)

    precision: an integer from 1 to 16 that can be set as the log grouping precision. A smaller value indicates a higher precision and more patterns. The default value is 3.

  - Returned fields:
    - pattern: the sub-patterns of log entries in a log group.
    - count: the number of log entries in a log group.
    - signature: the log pattern of a log group.
    - origin_signatures: the original signature of a log group. You can use this field to query raw log entries of the log group.
- Compare the number of log entries grouped in different time ranges.

- SQL statement:

```
* | select v.pattern, v.signature, v.count, v.count_compare, v.diff from (select compare_log_reduce(3, 86400) as v from log) order by v.diff desc limit 1000
```

> **Note** After you click **Log Compare** in the Log Service console, you can click **Copy Query** to obtain the SQL statement.

- Input parameters: compare_log_reduce(precision, compare_interval)
  - precision: an integer from 1 to 16 that can be set as the log grouping precision. A smaller value indicates a higher precision and more patterns. The default value is 3.
  - compare_interval: the number of seconds between when the previous log entries and the current log entries were generated. The value of this parameter must be a positive integer.
- Returned fields:
  - pattern: the sub-patterns of log entries in a log group.
  - signature: the log pattern of a log group.
  - count: the number of log entries in a log group.
  - count_compare: the number of log entries for a same-pattern log group within the specified time range.
  - Diff: the difference between the values of the count field and the count_compare field.

28.1.4.7.4. Contextual query

This topic describes the contextual query feature provided in the Log Service console. You can use this feature to query the full context of the log file where specified log entries are obtained.

Prerequisites

The index feature is enabled.

Context

The contextual query feature identifies the server and file where a specified log entry resides. It then queries several log entries before and after the log entry in the original log file. This helps you locate errors during troubleshooting.

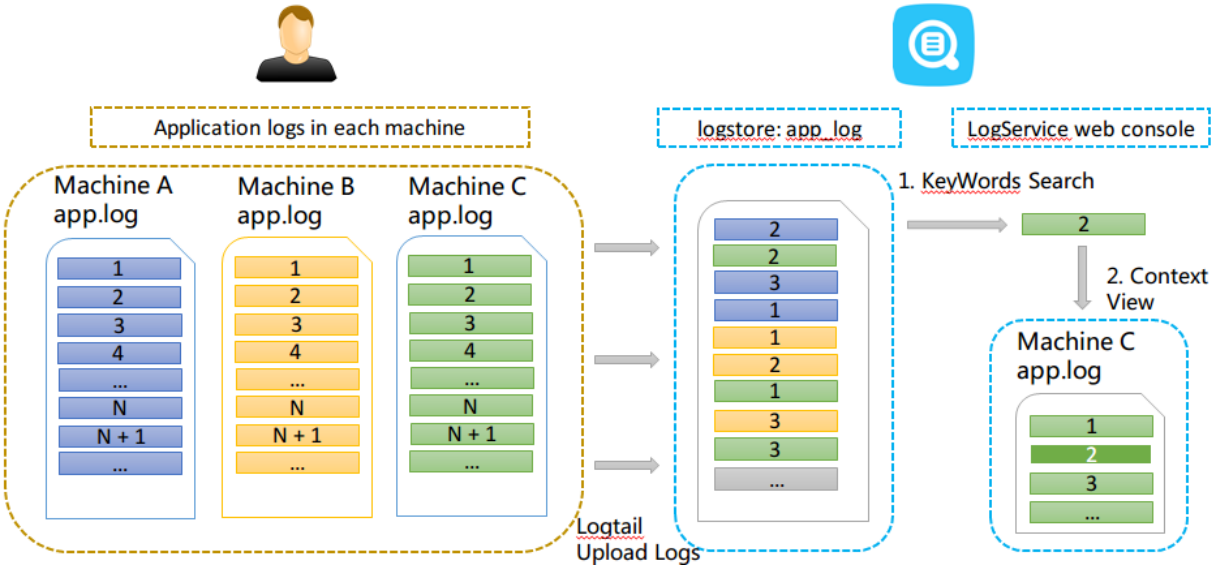
Scenarios

For example, a transaction on an O2O takeout website is logged in an application log file on a server as follows: **User logon** > **Browse products** > **Select an item** > **Add to a shopping cart** > **Place an order** > **Payment** > **Deduction** > **Generate an order**.

If the order fails, the O&M personnel must locate the cause at the earliest opportunity. In traditional contextual queries, the O&M personnel must be authorized before logging on to each server where the application is deployed. Then, the O&M personnel must use the order ID as a keyword to search application log files to locate the cause of the failure.

In Log Service, the O&M personnel can perform the following steps to locate the cause of the failure:

1. Install Logtail on servers. Then log on to the Log Service console to add the servers to machine groups and configure log collection. After the configurations are complete, Logtail starts to upload incremental logs.
2. On the search and analysis page of the Log Service console, specify the time range and find the error log entry based on the order ID.
3. Based on the error log entry, page up until you find log entries related to the error log entry. For example, you may want to find a log entry that records a credit card payment failure.



Note The contextual query feature does not support syslog.

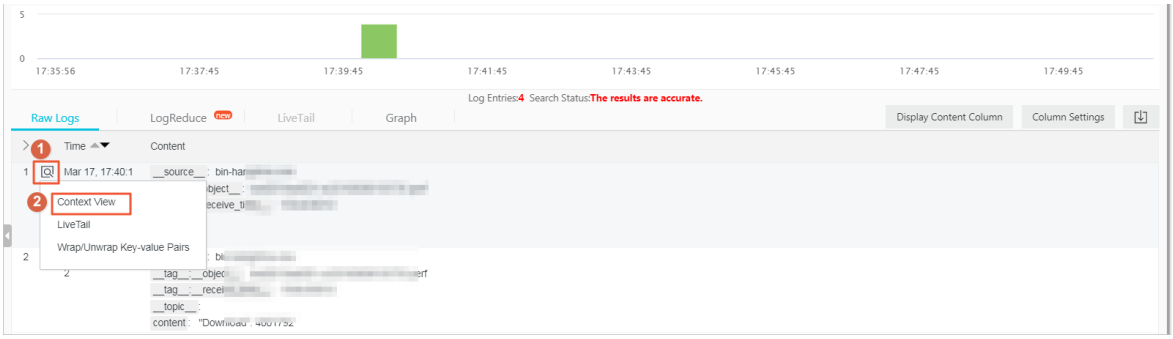
Benefits

- Intrusions into applications or changes to log file formats are avoided.
- You can view contextual log entries of a specified log entry in a log file on a server that has been registered in the Log Service console. This helps you avoid logging on to each server to search for logs that you want.
- You can specify the time range based on the time when an event occurs to locate suspicious log entries. Then you can perform a contextual query in the Log Service console. This improves troubleshooting efficiency.
- Data loss caused by log file rotation or insufficient storage space is avoided. You can view historical log data in the Log Service console at any time.

Procedure

1. Log on to the Log Service console.
2. Click the target project in the Projects section.
3. Click the icon next to the name of the Logstore, and then select **Search & Analysis**.
4. Enter a search and analytic statement, select a time range, and then click **Search & Analytics**.

   On the query results page, if the **Context View** icon is available in the drop-down list of the icon to the left of a log entry, the log entry supports contextual query.



5. Click the icon to the left of a log entry, and select **Context View** from the drop-down list. On the page that appears, view the contextual log entries of the selected log entry.
6. Scroll up and down to view more contextual log entries. To view earlier or later contextual log entries, click

Old or **New**.

28.1.4.7.5. Saved search

This topic describes how to save a search and analytic statement as a saved search for a Logstore. The saved search feature allows you to search and analyze log data in an efficient way.

Prerequisites

The index feature is enabled and configured.

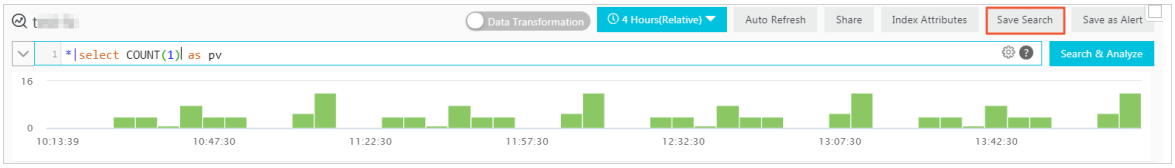
Context

If you need to frequently run a search and analytic statement, you can save the statement as a saved search. In later queries, you can click the name of the saved search on the left side of the search page to run the statement and view the result. You can also use the saved search in alert configurations. Log Service periodically runs the search and analytic statement and sends an alert if a query result meets the trigger condition.

If you want to select **Open Saved Search** in the Event Action field when you configure drill-down analysis, you must preset a saved search and set a **placeholder** in the query statement. For more information, see Drill-down analysis.

Procedure

1. Log on to the Log Service console.
2. Click the icon next to the name of the Logstore, and then select **Search & Analysis**.
3. Enter a search and analytic statement in the search box, set a time range, and then click **Search & Analyze**.
4. Click **Save Search** in the upper-right corner of the page.



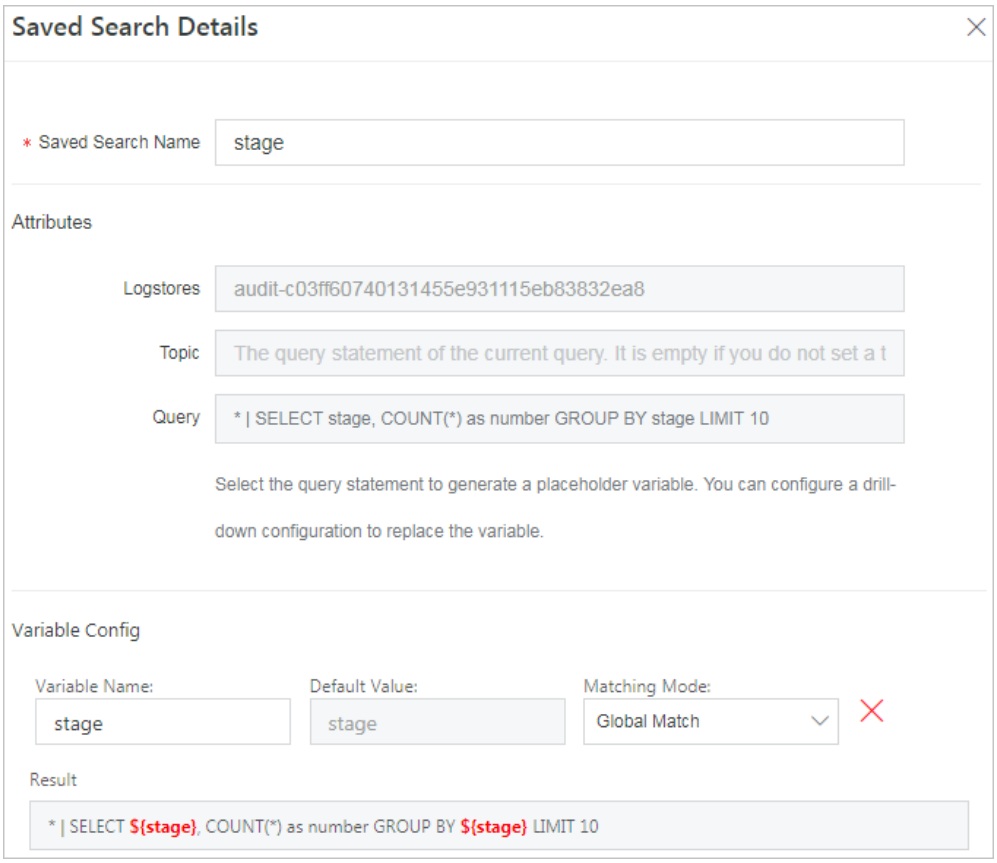
5. Configure the saved search.
   i. Enter a **Saved Search Name**.
      - The name can contain only lowercase letters, digits, hyphens (-), and underscores (_).
      - The name must start and end with a lowercase letter or digit.
      - The name must be 3 to 63 characters in length.
   ii. Check the values of the **Logstores**, **Topic**, and **Query** parameters.

      If the values of the **Logstores** and **Topic** parameters do not meet your requirements, follow these steps: Return to the Logstores page. On this page, find and click the name of the target Logstore. On the page that appears, enter the search and analytic statement in the Search & Analyze search box, and then click **Save Search** again.

iii. (Optional)Select a part of the query statement, and then click **Generate Variable**.

The generated variable is a placeholder variable. You can set the placeholder name in the **Variable Name** field. The selected characters are displayed in the **Default Value** field.

> **Note** If you use this saved search for drill-down analysis in another chart where the **variable** is the same as the **Variable Name**, the **Default Value** is replaced with the chart value that you click to trigger the drill-down event. The search and analytic statement with the replaced chart value is executed. For more information, see Drill-down analysis.

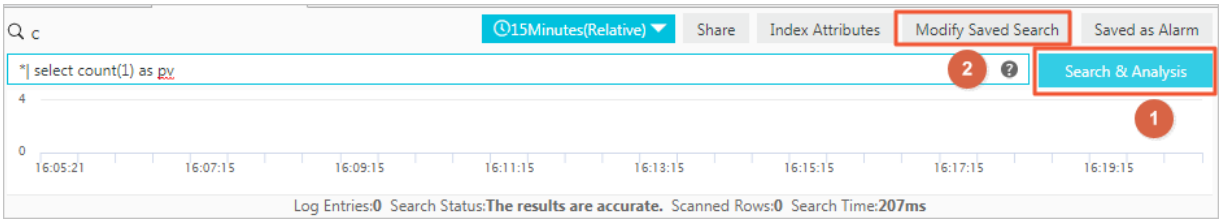


6. Click **OK**.

What's next

To modify a saved search, perform the following operations:

Enter a new search and analytic statement, click **Search & Analytics** to run the statement, and then click **Modify Saved Search**.



28.1.4.7.6. Quick analysis

This topic describes the quick analysis feature of Log Service. You can use this feature to query log data with one click. This feature allows you to analyze the distribution of a field in a specified time range and reduce the query cost of key data.

Features

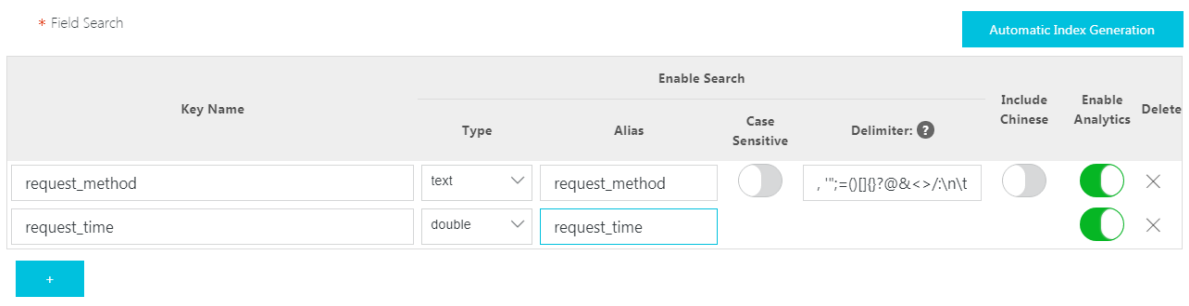
- Groups the first 100,000 values of a `text` field and provides statistics for the top 10 groups.
- Generates `approx_distinct` statements for `text` fields.
- Allows you to perform histogram-based statistics for the approximate distribution of `long` or `double` fields.
- Allows you to search for the maximum, minimum, and average of `long` or `double` fields and calculate the sum of the fields.
- Generates a query statement based on the quick analysis feature.

Prerequisites

Field indexes are configured.

- Indexes are configured for the fields that you need to search and analyze. For more information about how to enable the indexing feature, see Enable the index feature and configure indexes for a Logstore.
- The name of a field is specified as the `key`. The data type, alias, and delimiter of the field are configured.

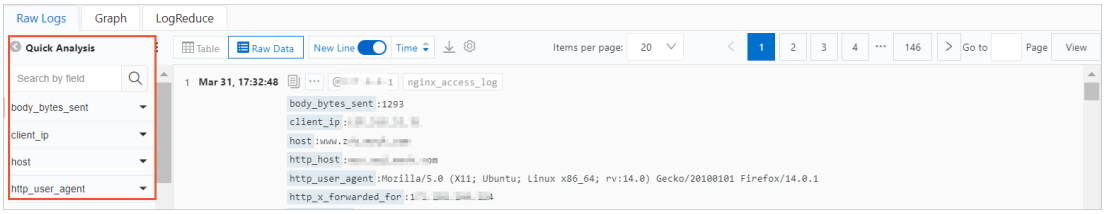
  For example, if a log entry contains the `request_method` and `request_time` fields, you can configure indexes for the two fields, as shown in the following figure.



Instructions

After you configure indexes for specified fields, you can go to the Search & Analysis page and click the **Raw Logs** tab to view the specified fields. The fields are listed in the **Quick Analysis** pane on the left of the raw log entries. You can click the icon above the serial number to hide the Quick Analysis pane. You can also click the **Eye** icon to perform quick analysis based on the **Current Time Zone** and **Current Search** conditions.

Quick analysis



Data of the text type

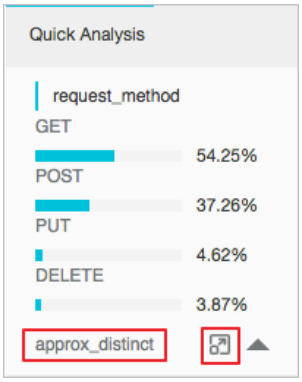
- Group and analyze log data

  Click the **Eye** icon next to a **text** field to group the first 100,000 field values and return the percentages of the top 10 groups.

  The following statement is used:

```
$Search | select ${keyName} , pv, pv *1.0/sum(pv) over() as percentage from( select count(1) as pv , "${keyName}" from (select "${keyName}" from log limit 100000) group by "${keyName}" order by pv desc) order by pv desc limit 10
```

The following figure shows the grouping and analytics result of the `request_method` field. `GET` requests account for the majority of requests.



- Calculate the number of unique values in a field

  Under the target fields in the **Quick Analysis** pane, click **Count Distinct Values** to calculate the number of unique values in the `${keyName}` field.

- Fill the Search & Analyze search box with the grouping and analytics statement

  Click the **Count Distinct Values** button on the right of the icon. The Search & Analyze search box is filled with the grouping and analytics statement. You can edit the statement.

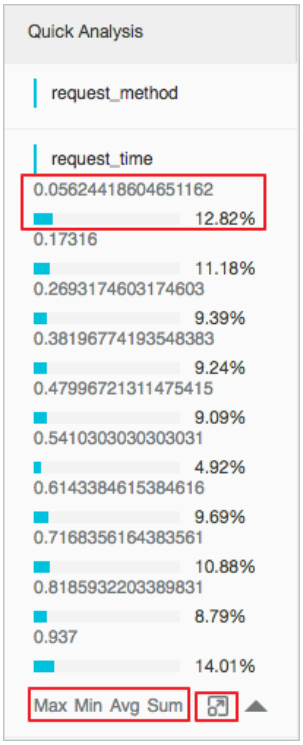
Data of the long and double types

- Display approximate distribution by using histograms

  The number of `long and double` field values is large. The preceding grouping and analytics method is not suitable for the long and double data types. You can use the following statement to divide field values into 10 groups and display the approximate distribution of the values in a histogram:

```
$Search | select numeric_histogram(10, ${keyName})
```

The following figure shows the approximate distribution of the values in the `request_time` field. The largest percentage of request time is distributed around 0.059 seconds.



- Perform quick analysis by using the `Max`, `Min`, `Avg`, and `Sum` functions

  You can click `Max` under a field to search for the maximum value, `Min` to search for the minimum value, `Avg` to calculate the average value, and `Sum` to calculate the sum of the values.

- Fill the Search & Analyze search box with the query statement of the histogram approximate distribution

  Click the icon next to `Sum`. The Search & Analyze search box is filled with the query statement of the histogram approximate distribution. You can edit the statement.

28.1.4.7.7. Other features

Log Service allows you to query log data by using various statements. It also provides many features for data search and analysis. This topic describes the raw log query, chart, saved search, dashboard, quick analysis, and alert features that Log Service supports.

Raw log query

After the index feature is enabled, enter one or more keywords in the search box and set the query time range. Then, click **Search & Analyze** to view the number of log entries displayed in a histogram, the raw logs, and the available charts.

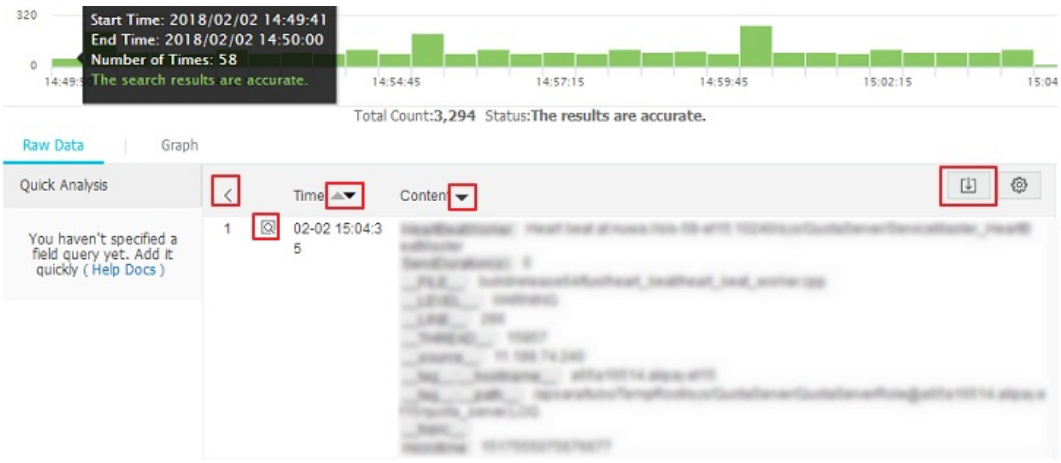
The histogram shows the time-based distribution of log entry occurrences. In the histogram, you can view the changes in the number of matched log entries over a period of time. You can click a rectangular to view the information about the occurrences within the specified time.

On the Raw Logs tab, you can view the matched log data in chronological order.

- You can click the triangle symbol next to **Time** to switch between the **chronological order** and **reverse chronological order**.
- You can click **Display Content Column** and then you can select **New Line** or **Full Line** to display the log entries. You can also set Truncate Character String.

- You can click the **keyword** of a field value in the log content to view all log entries that contain this keyword.
- You can click the icon in the upper-right corner of the Raw Logs tab to download the query results in the CSV format. You can also click **Column Settings** to select fields from the displayed raw log entries. This provides you with an easier way to view target field values of each raw log entry in new columns.
- You can click **Context View** to view the 15 log entries before and after the current log entry in a log file. For more information, see Contextual query.

> **Note** The contextual query feature applies only to the data that is uploaded by using Logtail.



Charts

After you have enabled the index feature and entered a search and analytic statement, you can select a chart on the **Graph** tab to visualize the query results.

- Multiple types of charts are provided, such as the table and line chart.

  You can select a chart type and configure the chart based on your business requirements.
- You can add charts to a **dashboard**. For more information, see Create and delete a dashboard.
- You can configure drill-down analysis for a chart. After you add a chart to a dashboard, you can click the chart data to trigger a drill down event and explore in-depth details about the data. For more information about how to configure drill-down analysis, see Drill-down analysis.

Dashboards

Log Service provides the dashboard feature that can visualize the results of search and analytic statements. For more information, see Create and delete a dashboard.

Alerts

Log Service allows you to configure alerts for your dashboard or the charts on your dashboard. You can configure rules to receive alerts from Log Service.

For more information, see Configure alerts.

28.1.4.8. SQL syntax and functions

28.1.4.8.1. General aggregate functions

This topic describes the syntax and examples of general aggregate functions.

The search and analytics feature of Log Service allows you to use general aggregate functions for log analysis. The following table describes the supported general aggregate functions.

Function	Description	Example
`arbitrary(x)`	Returns a random value from among the values in the x field.	`latency > 100 | select arbitrary(method)`
`avg(x)`	Returns the arithmetic mean of all values in the x field.	`latency > 100 | select avg(latency)`
`checksum(x)`	Returns a Base64-encoded checksum of the values of the x field.	`latency > 100 | select checksum(method)`
`count(*)`	Calculates the number of values of a field.	-
`count(x)`	Counts the number of non-null values of the x field.	`latency > 100 | select count(method)`
`count(digit)`	Counts the number of values of a field. The `count(digit)` function is equivalent to the `count(1)` and `count(*)` functions.	-
`count_if(x)`	Returns the number of the occurrences of the TRUE value.	`latency > 100 | select count_if(url like '%abc')`
`geometric_mean(x)`	Returns the geometric mean of the values in the x field.	`latency > 100 | select geometric_mean(latency)`
`max_by(x,y)`	Returns the value of the x field that corresponds to the maximum value of the y field.	`* | select min_by(method,latency,3)`: queries the method that corresponds to the maximum latency.
`max_by(x,y,n)`	Returns n values of the x field that corresponds to the n largest value of the y field.	`* | select min_by(method,latency,3)`: queries the three methods that corresponds to the three maximum latencies.
`min_by(x,y)`	Returns the value of the x field that corresponds to the smallest value of the y field.	`* | select min_by(x,y)`: queries the method that corresponds to the minimum latency.
`min_by(x,y,n)`	Returns n values of the x field that corresponds to the n smallest values of the y field.	`* | select min_by(method,latency,3)`: queries the three methods that corresponds to the three minimum latencies.
`max(x)`	Returns the maximum value among all values in the x field.	`latency > 100| select max(inflow)`
`min(x)`	Returns the minimum value among all values in the x field.	`latency > 100| select min(inflow)`
`sum(x)`	Returns the sum among all values in the x field.	`latency > 10 | select sum(inflow)`

Function	Description	Example
`bitwise_and_agg(x)`	Returns the bitwise AND of all values in the x field in two's complement representation.	-
`bitwise_or_agg(x)`	Returns the bitwise OR of all values in the x field in two's complement representation.	-

28.1.4.8.2. Security check functions

Security check functions in Log Services are designed based on the globally shared WhiteHat Security asset library. This topic describes security check functions that you can use to check whether an IP address, domain name, or URL in logs is secure.

Scenarios

- O&M personnel of enterprises and institutions in Internet, gaming, information, and other industries that require robust O&M services can use security check functions to identify suspicious requests or attacks. They can also use the functions to implement in-depth analysis and defend against potential attacks.
- O&M personnel of enterprises and institutions in banking, securities, e-commerce, and other industries that require strong protection for internal assets can use security check functions to identify requests to suspicious websites and downloads initiated by trojans. Then the O&M personnel can take immediate actions to prevent potential losses.

Features

- Reliability: built upon the globally shared WhiteHat Security asset library that is updated in a timely manner.
- Efficiency: capable of screening millions of IP addresses, domain names, and URLs within seconds.
- Ease of use: supports the analysis of network logs by using the security_check_ip, security_check_domain, and security_check_url functions.
- Flexibility: supports interactive queries, report creation, and alert configurations and subsequent actions.

Functions

Function	Description	Example
security_check_ip	Checks whether an IP address is secure. • The value 1 indicates that the specified IP address is suspicious. • The value 0 indicates that the specified IP address is secure.	`select security_check_ip(real_client_ip)`
security_check_domain	Checks whether a domain name is secure. • The value 1 indicates that the specified domain name is suspicious. • The value 0 indicates that the specified domain name is secure.	`select security_check_domain(site)`

Function	Description	Example
security_check_url	Checks whether a URL is secure. • The value 1 indicates that the specified URL is suspicious. • The value 0 indicates that the specified URL is secure.	`select security_check_domain(concat(host, url))`

Examples

- Check external suspicious requests and generate reports

  For example, an e-commerce enterprise collects logs from its NGINX servers and wants to scan suspicious client IP addresses. To do this, the enterprise can pass the ClientIP field in logs that are collected from the NGINX servers to the `security_check_ip` function and filter out IP addresses associated with the returned value 1. Then the enterprise can query the countries where the IP addresses are located and ISPs to which the IP addresses belong.

  SQL statement for this scenario:

  ```
  * | select ClientIP, ip_to_country(ClientIP) as country, ip_to_provider(ClientIP) as provider, count(1) as PV where security_check_ip(ClientIP) = 1 group by ClientIP order by PV desc
  ```

  Display the ISPs and countries in a map.

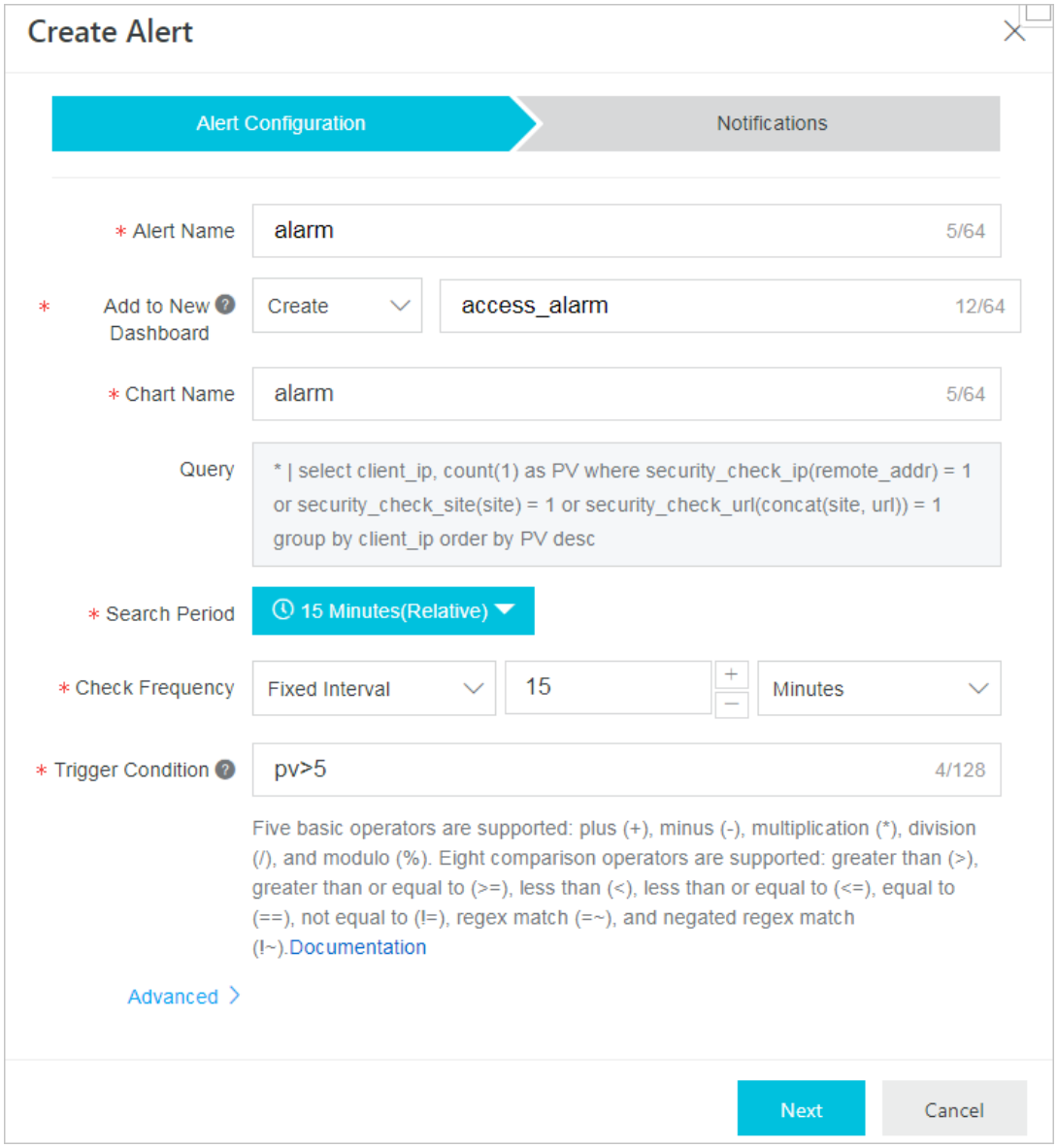
client_ip	country	provider	PV
180 3	CN		3
103	CN		3
180 7	CN		1

- Check internal suspicious requests and send alerts

  For example, a securities operator collects logs of its internal devices that access the Internet through gateways. To check requests to suspicious websites, the operator can run the following statement:

  ```
  * | select client_ip, count(1) as PV where security_check_ip(remote_addr) = 1 or security_check_site(site) = 1 or security_check_url(concat(site, url)) = 1 group by client_ip order by PV desc
  ```

  The operator can save this statement as a saved search and configure an alert. An alert is triggered when a client frequently accesses suspicious websites. The statement in the alert can be configured to run every five minutes to check if a client has frequently (more than five times) accessed suspicious websites in the past one hour. The following figure shows the configurations of an alert.



28.1.4.8.3. Map functions

This topic describes the syntax and examples of map functions.

The following table describes the supported map functions.

Function	Description	Example
Subscript operator []	Retrieves the value corresponding to a specified key from a map.	-

Function	Description	Example
histogram(x)	Groups the values of the parameter x and calculates the number of occurrences of each value. The syntax is equivalent to `select count group by x`. **Note** The returned data is in the JSON format.	The statement `latency > 10 | select histogram(status)` is equivalent to the statement `latency > 10 | select count(1) group by status`.
histogram_u(x)	Groups the values of the parameter x and calculates the number of occurrences of each value. **Note** The returned data is in the format of multiple rows and columns.	The statement `latency > 10 | select histogram_u(status)` is equivalent to the statement `latency > 10 | select count(1) group by status`.
map_agg(Key,Value)	Returns a random value of the key in the format of a map that consists of key-value pairs.	`latency > 100 | select map_agg(method,latency)`
multimap_agg(Key,Value)	Returns all values of the key in the format of a map that consists of key-value pairs.	`latency > 100 | select multimap_agg(method,latency)`
cardinality(x) → bigint	Returns the cardinality of the map x.	-
element_at(map `<K, V>`, key) → V	Returns the value for the specified key.	-
map() → map `<unknown, unknown>`	Returns an empty map.	-
map(array `<K>`, array `<V>`) → map `<K,V>`	Returns a map where each key-value pair consists of two elements from two separate arrays.	`SELECT map(ARRAY[1,3], ARRAY[2,4]); - {1 -> 2, 3 -> 4}`
map_from_entries(array `<row<K, V>>`) → map `<K,V>`	Converts a multi-dimensional array to a map.	`SELECT map_from_entries(ARRAY[(1, 'x'), (2, 'y')]); - {1 -> 'x', 2 -> 'y'}`
map_concat(map1 `<K, V>`, map2 `<K, V>`, ..., mapN `<K, V>`) → map `<K,V>`	Returns a map that is the union of all specified maps. If a key is found in multiple specified maps, the value of the key in the returned map is the value of the key that occurs in the last specified map.	-
map_filter(map `<K, V>`, function) → map `<K,V>`	For more information, see the map_filter function in Lambda functions.	-

Function	Description	Example
map_keys(x <K, V>) → array <K>	Returns an array of keys in the specified map.	-
map_values(x <K, V>) → array <V>	Returns an array of values in the specified map.	-

28.1.4.8.4. Approximate functions

This topic describes the syntax and examples of approximate functions that Log Service supports for log analysis.

The following table describes the supported approximate functions.

Function	Description	Example
`approx_distinct(x)`	Returns the approximate number of distinct values of the x field.	-
`approx_percentile(x,percentage)`	Returns the value located at the specified approximate percentage among the sorted values of the x field.	`approx_percentile(x,0.5)`: returns the median among the sorted values of the x field.
`approx_percentile(x,percentages)`	Returns values located at multiple specified approximate percentages among the sorted values of the x field. This function works in a similar manner to the `approx_percentile(x,percentage)` function.	`approx_percentile(x,array[0.1,0.2])`
`numeric_histogram(buckets,Value)`	Distributes all values of the *Value* field into multiple buckets. The *buckets* parameter specifies the number of buckets. The key of every bucket and the number of values in a bucket are returned. This function is equivalent to `select count group by`. **Note** The response is in the JSON format.	`method:POST | select numeric_histogram(10,latency)`: distributes the latencies of POST requests into 10 buckets and calculates the number of latencies in each bucket.

Function	Description	Example
`numeric_histogram_u(buckets, Value)`	Distribute all values of the *Value* field into multiple buckets. The *buckets* parameter specifies the number of buckets. The key of every bucket and the number of values in a bucket are returned. This function is equivalent to `select count group by` . **Note** The returned data is in the format of multiple rows and columns.	`method:POST | select numeric_histogram(10,latency)` : distributes the latency data of POST requests into 10 buckets and calculates the number of latency data in each bucket.

Note The number of values in every bucket is evenly distributed. It is returned along with the average value of all values in a bucket.

28.1.4.8.5. Mathematical statistics functions

This topic describes the syntax and examples of mathematical statistics functions.

The search and analytics feature of Log Service allows you to use mathematical statistics functions for log analysis. The following table describes the supported mathematical statistics functions.

Function	Description	Example
`corr(y, x)`	Returns the correlation coefficient of input values. The result ranges from 0 to 1.	`latency>100| select corr(latency,request_size)`
`covar_pop(y, x)`	Returns the population covariance of input values.	`latency>100| select covar_pop(request_size,latency)`
`covar_samp(y, x)`	Returns the sample covariance of input values.	`latency>100| select covar_samp(request_size,latency)`
`regr_intercept(y, x)`	Returns the linear regression intercept of input values. y is the dependent value. x is the independent value.	`latency>100| select regr_intercept(request_size,latency)`
`regr_slope(y,x)`	Returns the linear regression slope of input values. y is the dependent value. x is the independent value.	`latency>100| select regr_slope(request_size,latency)`
`stddev(x)` or `stddev_samp(x)`	Returns the sample standard deviation of the values in the x field.	`latency>100| select stddev(latency)`
`stddev_pop(x)`	Returns the population standard deviation of the values in the x field.	`latency>100| select stddev_pop(latency)`

Function	Description	Example
`variance(x)` or `var_samp(x)`	Returns the sample variance of the values in the x field.	`latency>100| select variance(latency)`
`var_pop(x)`	Returns the population variance of the values in the x field.	`latency>100| select variance(latency)`

28.1.4.8.6. Mathematical calculation functions

This topic describes the syntax and examples of mathematical calculation functions.

By including mathematical calculation functions in SQL statements, you can perform mathematical calculation on log query results.

Mathematical operators

Mathematical operators include the plus sign (+), minus sign (-), multiplication sign (*), division sign (/), and percent sign (%). These operators can be used in SELECT statements.

Example:

```
*|select avg(latency)/100 , sum(latency)/count(1)
```

Mathematical calculation functions

Log Service supports the following mathematical calculation functions.

Function	Description
`abs(x)`	Returns the absolute values of the values in the x field.
`cbrt(x)`	Returns the cube roots of the values in the x field.
`ceiling(x)`	Returns the rounded-up nearest integers of the values in the x field.
`cosine_similarity(x,y)`	Returns the cosine similarity between the sparse vectors x and y.
`degrees`	Converts angles in radians to degrees.
`e()`	Returns the Euler's number.
`exp(x)`	Returns Euler's number raised to the power of the values in the x field.
`floor(x)`	Returns the rounded-down nearest integers of the values in the x field.
`from_base(string,radix)`	Returns the radix number representation of a string.
`ln(x)`	Returns the natural logarithm of x.
`log2(x)`	Returns the base 2 logarithm of x.
`log10(x)`	Returns the base 10 logarithm of x.

Function	Description
`log(x,b)`	Returns the base b logarithm of x.
`pi()`	Returns the constant Pi.
`pow(x,b)`	Returns x raised to the power of b.
`radians(x)`	Converts angle x in degrees to radians.
`rand()`	Returns a random number.
`random(0,n)`	Returns a random number from 0 to n (exclusive).
`round(x)`	Returns x rounded to the nearest integer.
`round(x,y)`	Returns x rounded to y decimal places. For example, round(1.012345,2) = 1.01.
`sqrt(x)`	Returns the square root of x.
`to_base(x,radix)`	Returns the radix number representation of x.
`truncate(x)`	Returns x rounded to integer by dropping digits after the decimal point.
`acos(x)`	Returns the arc cosine of x.
`asin(x)`	Returns the arc sine of x.
`atan(x)`	Returns the arc tangent of x.
`atan2(y,x)`	Returns the arc tangent of y/x.
`cos(x)`	Returns the cosine of x.
`sin(x)`	Returns the sine of x.
`cosh(x)`	Returns the hyperbolic cosine of x.
`tan(x)`	Returns the tangent of x.
`tanh(x)`	Returns the hyperbolic tangent of x.
`infinity()`	Returns the value representing positive infinity.
`is_infinity(x)`	Determines whether x is infinite.
`is_finity(x)`	Determines whether x is finite.
`is_nan(x)`	Determines whether x is not-a-number.

28.1.4.8.7. String functions

This topic describes string functions that Log Service supports for log data search and analytics.

This following table lists the functions and their descriptions.

Function	Description
`chr(x)`	Converts an integer to an ASCII character. For example, the result of `chr(65)` is `A`.
`codepoint (x)`	Converts an ASCII character to a code point of the integer type. For example, the result of `codepoint('A')` is `65`.
`length(x)`	Returns the length of a string.
`levenshtein_distance(string1, string2)`	Returns the Levenshtein distance of string1 and string2.
`lower(string)`	Converts all uppercase characters in a string into lowercase characters.
`lpad(string, size, padstring)`	Pads a string to the specified size. If the length of the string is shorter than the specified size, padstring is used to left pad the string. If the length of the string is longer than the specified size, the string is truncated with the specified size.
`rpad(string, size, padstring)`	Right pads a `string` with the specified padding. The implementation is similar to that of the lpad function.
`ltrim(string)`	Removes whitespace characters from the left side of a string.
`replace(string, search)`	Removes all occurrences of a substring search from a string.
`replace(string, search,rep)`	Replaces all occurrences of a substring search in a string with another substring rep.
`reverse(string)`	Reverses a string.
`rtrim(string)`	Removes whitespace characters from the right side of a string.
`split(string,delimeter,limit)`	Splits a string based on the specified delimiter and returns an array with the maximum number of elements at limit. The index of the first element in the array is 1.
`split_part(string,delimeter,offset)`	Splits a string based on a delimiter into an array of substrings and returns the element with the index specified by the offset parameter.
`split_to_map(string, entryDelimiter, keyValueDelimiter) → map<varchar, varchar>`	Splits a string based on the entryDelemiter into multiple entries, each of which is then split based on the keyValueDelimiter into a key and value. This function returns a map.
`position(substring IN string)`	Returns the position of the first occurrence of the specified substring in a string.
`strpos(string, substring)`	Returns the position of the first occurrence of the specified substring in a string. Positions start with 1. If the substring is found, 0 is returned.

Function	Description
`substr(string, start)`	Returns a substring from the start position. Positions start with 1.
`substr(string, start, length)`	Returns a substring of a specified length from start position. Positions start with 1. The length parameter specifies the length of the substring returned.
`trim(string)`	Removes leading and trailing whitespace characters from a string.
`upper(string)`	Converts all lowercase characters in a string into uppercase characters.
`concat(string,string...)`	Concatenates multiple strings into a single string.
`hamming_distance (string1,string2)`	Returns the Hamming distance of string1 and string2.

Note Strings must be enclosed in single quotation marks ('). Double quotation marks (") are used to enclose field names. For example, `a='abc'` indicates `field a = string 'abc'`, and `"a"="abc"` indicates `field a = field abc`.

28.1.4.8.8. Date and time functions

This topic describes the available time functions, date functions, interval functions, and a time series padding function in Log Service. You can use these functions when you analyze data.

Date and time data types

- Unix timestamp: specifies the number of seconds that have elapsed since 1970-01-01 00:00:00 UTC. The value is in the format of an integer. For example, `1512374067` indicates `Mon Dec 4 15:54:27 CST 2017`. The `__time__` field in every log entry is of this type.
- Timestamp: specifies the date and time in the format of a string. For example, `2017-11-01 13:30:00`.

Date functions

The following table lists the commonly used date functions that are supported in Log Service.

Function	Description	Example
`current_date`	Returns the current date.	`latency>100| select current_date`
`current_time`	Returns the current time.	`latency>100| select current_time`
`current_timestamp`	Returns the current timestamp. This function is equivalent to the combination of the current_date and current_time functions.	`latency>100| select current_timestamp`
`current_timezone()`	Returns the current time zone.	`latency>100| select current_timezone()`

Function	Description	Example
`from_iso8601_timestamp(string)`	Parses an ISO 8601 formatted string into a timestamp that specifies the time zone.	`latency>100| select from_iso8601_timestamp(iso8601)`
`from_iso8601_date(string)`	Parses an ISO 8601 formatted string into a date.	`latency>100| select from_iso8601_date(iso8601)`
`from_unixtime(unixtime)`	Parses a Unix timestamp into a timestamp.	`latency>100| select from_unixtime(1494985275)`
`from_unixtime(unixtime,string)`	Parses a Unix timestamp into a timestamp based on the time zone that is specified by the string parameter.	`latency>100| select from_unixtime (1494985275,'Asia/Shanghai')`
`localtime`	Returns the local time.	`latency>100| select localtime`
`localtimestamp`	Returns the local timestamp.	`latency>100| select localtimestamp`
`now()`	Returns the current date and time. This function is equivalent to the `current_timestamp` function.	N/A
`to_unixtime(timestamp)`	Parses a timestamp into a Unix timestamp.	`*| select to_unixtime('2017-05-17 09:45:00.848 Asia/Shanghai')`

Time functions

The following table lists the time functions that you can use in Log Service to parse time in the formats supported in MySQL, such as %a, %b, and %y.

Function	Description	Example
`date_format(timestamp, format)`	Formats a timestamp in the specified format.	`latency>100 | select date_format (date_parse('2017-05-17 09:45:00','%Y-%m-%d %H:%i:%S'), '%Y-%m-%d')`
`date_parse(string, format)`	Parses a formatted string into a timestamp.	`latency>100 | select date_format (date_parse(time,'%Y-%m-%d %H:%i:%S'), '%Y-%m-%d')`

Formats

Format	Description
%a	The week day in abbreviation, such as Sun and Sat.
%b	The month in abbreviation, such as Jan and Dec.
%c	The month in the numeric format. Valid values: 1 to 12.

Format	Description
%D	The day of the month with a suffix, such as 0th, 1st, 2nd, and 3rd.
%d	The day of the month in the numeric format. Valid values: 01 to 31.
%e	The day of the month in the numeric format. Valid values: 1 to 31.
%H	The hour that applies the 24-hour clock convention. Valid values: 00 to 23.
%h	The hour that applies the 12-hour clock convention. Valid values: 01 to 12.
%I	The hour that applies the 12-hour clock convention. Valid values: 1 to 12.
%i	The minute in the numeric format. Valid values: 00 to 59.
%j	The day of the year. Valid values: 001 to 366.
%k	The hour. Valid values: 0 to 23.
%l	The hour. Valid values: 1 to 12.
%M	The month. Valid values: January to December.
%m	The month in the numeric format. Valid values: 01 to 12.
%p	The abbreviation that indicates the morning or afternoon. Valid values: a.m. and p.m..
%r	The time that applies the 12-hour clock convention: `hh:mm:ss AM/PM`.
%S	The second. Valid values: 00 to 59.
%s	The second. Valid values: 00 to 59.
%T	The time that applies the 24-hour clock convention, formatted in `hh:mm:ss`.
%U	The week of the year. Sunday is the first day of a week. Valid values: 00 to 53.
%u	The week of the year. Monday is the first day of a week. Valid values: 00 to 53.
%V	The week of the year. Sunday is the first day of a week. This format is used together with %X. Valid values: 01 to 53.
%v	The week of the year. Monday is the first day of a week. This format is used in combination with %x. Valid values: 01 to 53.
%W	The day of the week. Valid values: Sunday to Saturday.
%w	The day of the week. Valid values: 0 to 6. The value 0 indicates Sunday.
%Y	The year in the 4-digit format.
%y	The year in the 2-digit format.
%%	Escapes the second percent sign (%).

Truncation functions

Log Service supports a truncation function, which can truncate a time by the second, minute, hour, day, month, or year. Typically, this function is used for time-based analytics.

- Syntax

```
date_trunc(unit, x)
```

- Parameters

  The value of the x parameter can be a timestamp or Unix timestamp.

  The following table lists the values of the unit parameter and the responses when the x parameter is set to `2001-08-22 03:04:05.000`.

Unit	Response
second	2001-08-22 03:04:05.000
minute	2001-08-22 03:04:00.000
hour	2001-08-22 03:00:00.000
day	2001-08-22 00:00:00.000
week	2001-08-20 00:00:00.000
month	2001-08-01 00:00:00.000
quarter	2001-07-01 00:00:00.000
year	2001-01-01 00:00:00.000

- Example

  The date_trunc function is applicable only to analytics at a fixed time interval. To implement analytics at a flexible interval, for example, every 5 minutes, you need to use a GROUP BY clause according to the mathematical modulus method.

```
* | SELECT count(1) as pv,  __time__ - __time__% 300 as minute5 group by minute5 limit 100
```

  In the preceding statement, `%300` indicates that modulo and truncation are performed every 5 minutes.

  The following example shows how to use the date_trunc function:

```
*|select  date_trunc('minute' ,  __time__)  as t,
       truncate (avg(latency) ) ,
       current_date
       group by   t
       order by t  desc
       limit 60
```

Interval functions

Interval functions perform interval-related calculation. For example, you can use interval functions to add or subtract an interval based on a date, or calculate the interval between two dates.

Function	Description	Example

Function	Description	Example
`date_add (unit, value, timestamp)`	Adds an interval `value` of the `unit` type to a `timestamp`. To subtract an interval, use a negative `value`.	`date_add('day', -7, '2018-08-09 00:00:00')`: indicates seven days before August 9.
`date_diff(unit, timestamp1, timestamp2)`	Returns the time difference between `timestamp1` and `timestamp2` expressed in terms of `unit`.	`date_diff('day', '2018-08-02 00:00:00', '2018-08-09 00:00:00') = 7`

The following table lists the values of the unit parameter that are supported by the interval functions.

Value	Description
millisecond	The millisecond.
second	The second.
minute	The minute.
hour	The hour.
day	The day.
week	The week.
month	The month.
quarter	The quarter, namely, three months.
year	The year.

Time series padding function

The time series padding function is used to pad time series and corresponding data.

> **Note** This function must be used in combination with the `group by time order by time` clause. When used together, the `order by` clause does not support the `desc` sorting method.

- Syntax

```
time_series(time_column, window, format, padding_data)
```

- Parameters

Parameter	Description
time_column	The name of the time field in a log entry. The default field name is `__time__`. The field value is of the LONG or TIMESTAMP type.
window	The time window for a data query. It is composed of a number and a unit. Unit: s (seconds), m (minutes), H (hours), and d (days). For example, 2h, 5m, or 3d.
format	The MySQL time format displayed.

Parameter	Description
padding_data	The content to be added for a time point. Valid values: ○ 0: adds 0. ○ null: adds null. ○ last: adds the value corresponding to the last time point. ○ next: adds the value corresponding to the next time point. ○ avg: adds the average value of the last and next values.

- Example

  The following statement is used to format data every 2 hours:

```
* | select time_series(__time__, '2h', '%Y-%m-%d %H:%i:%s', '0')  as stamp, count(*) as num from log group by stamp order by stamp
```

28.1.4.8.9. URL functions

This topic describes the syntax of URL functions and provides examples.

URL functions extract fields from standard URLs. A standard URL is described as follows:

```
[protocol:][//host[:port]][path][? query][#fragment]
```

Common URL functions

Function	Description	Example	
		Statement	Response
`url_extract_fragment(url)`	Extracts the fragment identifier from a URL and returns the fragment identifier of the VARCHAR type.	`*| select url_extract_fragment('https://sls.console.aliyun.com/#/project/dashboard-demo/categoryList')`	`/project/dashboard-demo/categoryList`
`url_extract_host(url)`	Extracts the host information from a URL and returns the host information of the VARCHAR type.	`*|select url_extract_host('http://www.aliyun.com/product/sls')`	`www.aliyun.com`
`url_extract_parameter(url, name)`	Extracts the value of the name parameter in the query from a URL and returns the value of the VARCHAR type.	`*|select url_extract_parameter('http://www.aliyun.com/product/sls?userid=testuser','userid')`	`testuser`
`url_extract_path(url)`	Extracts the path from a URL and returns the path of the VARCHAR type.	`*|select url_extract_path('http://www.aliyun.com/product/sls?userid=testuser')`	`/product/sls`
`url_extract_port(url)`	Extracts the port number from a URL and returns the port number of the BIGINT type.	`*|select url_extract_port('http://www.aliyun.com:80/product/sls?userid=testuser')`	`80`

Function	Description	Example	
		Statement	Response
`url_extract_protocol(url)`	Extracts the protocol from a URL and returns the protocol of the VARCHAR type.	`*|select url_extract_protocol('http://www.aliyun.com:80/product/sls?userid=testuser')`	`http`
`url_extract_query(url)`	Extracts the query string from a URL and returns the query string of the VARCHAR type.	`*|select url_extract_query('http://www.aliyun.com:80/product/sls?userid=testuser')`	`userid=testuser`
`url_encode(value)`	Encodes a URL.	`*|select url_encode('http://www.aliyun.com:80/product/sls?userid=testuser')`	`http%3a%2f%2fwww.aliyun.com%3a80%2fproduct%2fsls%3fuserid%3dtestuser`
`url_decode(value)`	Decodes a URL.	`*|select url_decode('http%3a%2f%2fwww.aliyun.com%3a80%2fproduct%2fsls%3fuserid%3dtestuser')`	`http://www.aliyun.com:80/product/sls?userid=testuser`

28.1.4.8.10. Regular expression functions

This topic describes the available regular expression functions. You can use these functions when you query and analyze data in Log Service.

A regular expression function parses a string and returns the required substrings.

The following table lists common regular expression functions.

Function	Description	Example
`regexp_extract_all(string, pattern)`	Returns an array where each element is a substring that matches the regular expression. These substrings derive from the specified string.	The returned result of `*| SELECT regexp_extract_all('5a 67b 890m', '\d+')` is `['5','67','890']`. The returned result of `*|SELECT regexp_extract_all('5a 67a 890m', '(\d+)a')` is `['5a','67a']`.
`regexp_extract_all(string, pattern, group)`	Returns an array where each element is a part of a substring that matches the regular expression. This part is the content in the group parameter value of the `()` of a substring that derives from the specified string.	The returned result of `*| SELECT regexp_extract_all('5a 67a 890m', '(\d+)a',1)` is `['5','67']`.
`regexp_extract(string, pattern)`	Returns the first substring of the specified string that matches the regular expression.	The returned result of `*|SELECT regexp_extract('5a 67b 890m', '\d+')` is `'5'`.

Function	Description	Example
`regexp_extract(string, pattern,group)`	Returns a part of the first substring that matches the regular expression. This part is the content in the group parameter value of the `()` of the substring that derives from the specified string.	The returned result of `*|SELECT regexp_extract('5a 67b 890m', '(\d+)([a-z]+)',2)` is `'a'`.
`regexp_like(string, pattern)`	Returns a Boolean value. If the string and its substrings cannot match the regular expression, the value False is returned.	The returned result of `*|SELECT regexp_like('5a 67b 890m', '\d+m')` is True.
`regexp_replace(string, pattern, replacement)`	Replaces the substrings of the specified string that match the regular expression with the value of the replacement parameter.	The returned result of `*|SELECT regexp_replace('5a 67b 890m', '\d+','a')` is `'aa ab am'`.
`regexp_replace(string, pattern)`	Removes the substrings of the specified string that match the regular expression. This function is equivalent to `regexp_replace(string,patterm,'')`.	The returned result of `*|SELECT regexp_replace('5a 67b 890m', '\d+')` is `'a b m'`.
`regexp_split(string, pattern)`	Returns an array where each element is a substring of the specified string that is split based on the regular expression.	The returned result of `*|SELECT regexp_split('5a 67b 890m', '\d+')` is `['a','b','m']`.

28.1.4.8.11. JSON functions

JSON functions can convert a string into a JSON type and extract JSON fields. The two major JSON data types are map and array. If a string cannot be converted to a value of the JSON type, null value is returned.

For information about how to expand JSON data into multiple rows, see UNNEST function.

The following table lists the JSON functions that Log Service supports:

Function	Description	Example
`json_parse(string)`	Converts a string to a JSON-formatted data.	`SELECT json_parse('[1, 2, 3]')`: returns a JSON array.
`json_format(json)`	Converts JSON-formatted data to a string.	`SELECT json_format(json_parse('[1, 2, 3]'))`: returns a string.
`json_array_contains(json, value)`	Determines whether a value exists in a JSON array or in a string that contains a JSON array.	`SELECT json_array_contains(json_parse('[1, 2, 3]'), 2)` or `SELECT json_array_contains('[1, 2, 3]', 2)`

Function	Description	Example
`json_array_get(json_array, index)`	Retrieves the element at the specified index in the JSON array. This function is equivalent to `json_array_contains`.	`SELECT json_array_get('["a", "b", "c"]', 0): returns 'a'`
`json_array_length(json)`	Returns the length of the JSON array.	`SELECT json_array_length('[1, 2, 3]'): returns 3`
`json_extract(json, json_path)`	Extracts a value from a JSON object and returns a JSON object. The json_path expression works in a similar manner to `$.store.book[0].title`.	`SELECT json_extract(json, '$.store.book')`
`json_extract_scalar(json, json_path)`	Returns a string. This function works in a similar manner to `json_extract`).	N/A
`json_size(json,json_path)`	Retrieves the length of a JSON object or array.	`SELECT json_size('[1, 2, 3]'): returns 3.`

28.1.4.8.12. Type conversion functions

Type conversion functions convert the data type of a specified value or column in a query.

You can use the index attribute feature of Log Service to set the data type of a field to LONG, DOUBLE, TEXT, or JSON. You can also query fields of various data types, including BIGINT, DOUBLE, VARCHAR, and TIMESTAMP. To query fields of a specific data type, you can use type conversion functions to convert the data type configured in an index into the data type that you use in a query.

Syntax

Note We recommend that you use the try_cast() function if a log contains dirty data. Otherwise, a query may fail due to the dirty data.

- Convert the data type of a column of values or a specific value into the specified type in a query. If the data type of a value fails to be converted, the query is terminated.

```
cast([key|value] AS type)
```

- Convert the data type of a column of values or a specific value into the specified type in a query. If the data type of a value fails to be converted, NULL is returned for the value, and the query continues.

```
try_cast([key|value] AS type)
```

Parameter	Description
key	The key of a field whose value data type is to be converted.
value	The field value whose data type is to be converted into the specified type.

Example

- To convert the numeric value 123 to a value of the VARCHAR type, use the following statement:

```
cast(123 AS varchar)
```

- To convert the data type of the uid field values to the VARCHAR type, use the following statement:

```
cast(uid AS varchar)
```

28.1.4.8.13. IP functions

This topic describes the syntax of IP functions and provides examples.

IP functions can identify whether an IP address is an intranet or Internet IP address. IP functions can also identify the country, province, and city where an IP address resides. For information about geohash functions, see Geography functions.

Function	Description	Example
`ip_to_domain(ip)`	Identifies whether an IP address is an intranet or Internet IP address. This function returns intranet or internet.	`SELECT ip_to_domain(ip)`
`ip_to_country(ip)`	Identifies the country where an IP address resides.	`SELECT ip_to_country(ip)`
`ip_to_province(ip)`	Identifies the province where an IP address resides.	`SELECT ip_to_province(ip)`
`ip_to_city(ip)`	Identifies the city where an IP address resides.	`SELECT ip_to_city(ip)`
`ip_to_geo(ip)`	Identifies the longitude and latitude of the city where an IP address resides. The result is returned in the format of `latitude, longitude`.	`SELECT ip_to_geo(ip)`
`ip_to_city_geo(ip)`	Identifies the longitude and latitude of the city where an IP address resides. Each city has only one longitude and latitude. The result is returned in the format of `latitude, longitude`.	`SELECT ip_to_city_geo(ip)`
`ip_to_provider(ip)`	Identifies the network service provider that assigns an IP address.	`SELECT ip_to_provider(ip)`
`ip_to_country(ip,'en')`	Identifies the country where an IP address resides. The function returns a country code.	`SELECT ip_to_country(ip,'en')`
`ip_to_country_code(ip)`	Identifies the country where an IP address resides. The function returns a country code.	`SELECT ip_to_country_code(ip)`
`ip_to_province(ip,'en')`	Identifies the province where an IP address resides.	`SELECT ip_to_province(ip,'en')`
`ip_to_city(ip,'en')`	Identifies the city where an IP address resides.	`SELECT ip_to_city(ip,'en')`

Example

- To query the number of requests that are not sent from an intranet, run the following statement:

```
* | SELECT count(1) where ip_to_domain(ip)! ='intranet'
```

- To query the top 10 provinces from which requests are sent, run the following statement:

```
* | SELECT count(1) as pv, ip_to_province(ip) as province GROUP BY province order by pv desc limit
10
```

Sample response

```
[
    {
        "__source__": "",
        "__time__": "1512353137",
        "province": "Zhejiang",
        "pv": "4045"
    }, {
        "__source__": "",
        "__time__": "1512353137",
        "province": "Shanghai",
        "pv": "3727"
    }, {
        "__source__": "",
        "__time__": "1512353137",
        "province": "Beijing",
        "pv": "954"
    }, {
        "__source__": "",
        "__time__": "1512353137",
        "province": "intranet IP address",
        "pv": "698"
    }, {
        "__source__": "",
        "__time__": "1512353137",
        "province": "Guangdong",
        "pv": "472"
    }, {
        "__source__": "",
        "__time__": "1512353137",
        "province": "Fujian",
        "pv": "71"
    }
]
```

The response contains an intranet IP address. You can use the SELECT statement to filter requests that are sent from the IP address.

- To query the top 10 provinces from which intranet requests are sent, run the following statement:

```
* | SELECT count(1) as pv, ip_to_province(ip) as province WHERE ip_to_domain(ip) ! = 'intranet'  G
ROUP BY province ORDER BY pv desc limit 10
```

- To query the average request latency, the maximum request latency, and the request with the maximum latency from each country, run the following statement:

```
* | SELECT AVG(latency),MAX(latency),MAX_BY(requestId, latency) ,ip_to_country(ip) as country grou
p by country limit 100
```

- To query the average latency of requests supported by each network service provider, run the following statement:

```
* | SELECT AVG(latency) , ip_to_provider(ip) as provider group by provider limit 100
```

- To query the longitude and latitude of the city to which an IP address belongs and show the city on a map, run the following statement:

```
* | SELECT count(1) as pv , ip_to_geo(ip) as geo group by geo order by pv desc
```

The following table shows the format of the result.

pv	geo
100	35.3284,-80.7459

28.1.4.8.14. GROUP BY syntax

This topic describes the GROUP BY syntax.

GROUP BY statements support multiple columns. A GROUP BY statement allows you to specify a column alias in a SELECT statement to as the corresponding KEY.

Example:

```
method:PostLogstoreLogs |select avg(latency),projectName,date_trunc('hour',__time__) as hour   group by projectName,hour
```

The hour alias indicates the third SELECT column `date_trunc('hour',__time__)`. This improves the performance of complicated queries.

The GROUP BY statement supports the GROUPING SETS, CUBE, and ROLLUP clauses.

Example:

```
method:PostLogstoreLogs |select avg(latency)   group by cube(projectName,logstore)
method:PostLogstoreLogs |select avg(latency)   group by GROUPING SETS ( ( projectName,logstore), (projectName,method))
method:PostLogstoreLogs |select avg(latency)   group by rollup(projectName,logstore)
```

Examples

- Use GROUP BY based on time

  Each log has a built-in time column named `__time__`. When the analytics feature is enabled on one of the columns, the statistics of the time column are included.

  The `date_trunc` function can truncate the time column to minute, hour, day, month, and year. The `date_trunc` function accepts an aligned unit and a column of the Unix timestamp type, such as `__time__`.

  - PV statistics per hour and per minute

```
* | SELECT count(1) as pv , date_trunc('hour',__time__) as hour group by hour order by hour limit 100
* | SELECT count(1) as pv , date_trunc('minute',__time__) as minute group by minute order by minute limit 100
```

> **Note** 100 specifies that up to 100 rows can be retrieved. If a LIMIT clause is not specified, up to 10 rows of data are retrieved by default.

- date_trunc functions are available only for statistics at a fixed time interval. For statistics based on flexible time dimensions, for example, every 5 minutes, run a GROUP BY statement based on the mathematical modulus method.

  ```
  * | SELECT count(1) as pv,  __time__ - __time__% 300 as minute5 group by minute5 limit 100
  ```

  In the preceding statement, `%300` indicates that the time is truncated in mod every 5 minutes.

- Retrieve non-aggregation columns from a GROUP BY statement

  In standard SQL, if a GROUP BY statement is used during the SELECT operation, Log Service selects only the raw data of the column specified in the GROUP BY statement or performs aggregation on any columns.

  For example, the following statement is invalid. Log Service cannot determine which row of b to return during the GROUP BY operation based on a, because b is not a GROUP BY column.

  ```
  *|select a, b , count(c) group by a
  ```

  Instead, you can use the `arbitrary()` function to return b.

  ```
  *|select a, arbitrary(b), count(c) group by a
  ```

28.1.4.8.15. Window functions

This topic describes the syntax for window functions.

Window functions are used to perform calculations across rows of a log. Common SQL aggregate functions calculate the results of only one row or aggregate all rows into one row for calculation. Window functions support cross-row calculation and fill the calculation results in each row.

Syntax for window functions is

```
SELECT key1, key2, value,
       rank() OVER (PARTITION BY key2
                    ORDER BY value DESC) AS rnk
FROM orders
ORDER BY key1,rnk
```

The important part is

```
rank() OVER (PARTITION BY  KEY1 ORDER BY KEY2 DESC)
```

rank() is an aggregate function. You can use any functions in analysis syntax or the function listed in this topic. PARTITION BY indicates the buckets based on which values are calculated.

Special aggregate functions used in windows

Function	Description
rank()	Returns the rank of a value within a group of values. The rank is one plus the number of preceding rows that are not peers of the current row.
row_number()	Returns a unique, sequential number for each row.
first_value(x)	Returns the first value of the window. In most cases, the function is used to obtain the maximum value after the values of the window are sorted.

Function	Description
last_value(x)	Returns the last value of the window. In most cases, the function is used to obtain the minimum value after the values of the window are sorted.
nth_value(x, offset)	Returns the value at the specified offset from the beginning of the window.
lead(x,offset,defaut_value)	Returns the value at offset rows after the current row in the window. If the target row does not exist, the default_value is returned.
lag(x,offset,defaut_value)	Returns the value at offset rows after the current row in the window. If the target row does not exist, the default_value is returned.

Example

- Rank the salaries of employees in their respective departments

```
* | select department, persionId, sallary , rank() over(PARTITION BY department order by sallary d
esc) as sallary_rank  order by department,sallary_rank
```

Results

department	persionId	sallary	sallary_rank
dev	john	9000	1
dev	Smith	8000	2
dev	Snow	7000	3
dev	Achilles	6000	4
Marketing	Blan Stark	9000	1
Marketing	Rob Stark	8000	2
Marketing	Sansa Stark	7000	3

- Calculate the salaries of employees as percentages in their respective departments

```
* | select department, persionId, sallary *1.0 / sum(sallary) over(PARTITION BY department ) as sa
llary_percentage
```

Results

department	persionId	sallary	sallary_percentage
dev	john	9,000	0.3
dev	Smith	8,000	0.26
dev	Snow	7000	0.23
dev	Achilles	6000	0.2

department	persionId	sallary	sallary_percentage
Marketing	Blan Stark	9000	0.375
Marketing	Rob Stark	8000	0.333
Marketing	Sansa Stark	7000	0.29

- Calculate the daily UV increase over the previous day

```
* | select day ,uv, uv *1.0 /(lag(uv,1,0) over() ) as diff_percentage from
(
select approx_distinct(ip) as uv, date_trunc('day',__time__) as day from log group by day order by
day asc
)
```

Results

day	uv	diff_percentage
2017-12-01 00:00:00	100	null
2017-12-02 00:00:00	125	1.25
2017-12-03 00:00:00	1,500	1.2
2017-12-04 00:00:00	175	1.16
2017-12-05 00:00:00	2,000	1.14
2017-12-06 00:00:00	225	1.125
2017-12-07 00:00:00	250	1.11

28.1.4.8.16. HAVING syntax

This topic describes the HAVING syntax.

The LogSearch/Analytics feature of Log Service supports the standard SQL HAVING clause. The HAVING clause is used with the GROUP BY statement to filter GROUP BY results.

Example:

```
method :PostLogstoreLogs |select avg(latency),projectName group by projectName  HAVING avg(latency) >
100
```

Difference between HAVING and WHERE clauses

The HAVING clause is used to filter the aggregation and calculation results after you run the GROUP BY statement. The WHERE clause is used to filter the raw data during the aggregation calculation.

Example

Calculate the average rainfall of each province where the temperature is higher than 10°C, and only show the provinces with average rainfall greater than 100 mL in the final results:

```
* | select avg(rain) ,province where temperature > 10 group by province having avg(rain) > 100
```

28.1.4.8.17. ORDER BY syntax

This topic describes the ORDER BY syntax.

The ORDER BY keyword is used to sort output results. You can sort output results based on only one column.

- Syntax format

```
order by column name [desc|asc]
```

- Example

```
method :PostLogstoreLogs |select avg(latency) as avg_latency,projectName group by projectName
HAVING avg(latency) > 5700000
order by avg_latency desc
```

28.1.4.8.18. LIMIT syntax

The LIMIT clause is used to limit the number of returned rows.

Syntax formats

Log Service supports the following LIMIT syntax formats:

- Reads only the first N rows:

```
limit N
```

- Reads N rows starting from the S-th row:

```
limit S , N
```

> **Note**
>
> - If you use the LIMIT clause to paginate results, the final results rather than the intermediate results of the SQL query are obtained.
> - You cannot apply the LIMIT clause to subqueries. For example, the following statement is not supported:
>
>   ```
>   * | select count(1) from ( select distinct(url) from limit 0,1000)
>   ```
>
> - If you use the LIMIT clause for pagination, the offset value cannot exceed 1,000,000. For example, in the `limit S , N` clause, the sum of S and N cannot exceed 1,000,000, and the value of N cannot exceed 10,000.

Example

- To obtain the first 100 rows of results, run the following statement.

```
* | select distinct(url) from log limit 100
```

- To obtain a total of 1,000 results from row 0 to row 999, run the following statement.

```
* | select distinct(url) from log limit 0,1000
```

- To obtain a total of 1,000 results from row 1,000 to row 1,999, run the following statement:

```
* | select distinct(url) from log limit 1000,1000
```

28.1.4.8.19. Syntax for CASE statements and if() functions

This topic describes the syntax for CASE statements and if() functions.

CASE statements are used to classify continuous data. For example, you can use the following CASE statement to extract information from http_user_agent and classify the information into two types: Android and iOS.

```
SELECT
 CASE
 WHEN http_user_agent like '%android%' then 'android'
 WHEN http_user_agent like '%ios%' then 'ios'
 ELSE 'unknown' END
 as http_user_agent,
    count(1) as pv
    group by  http_user_agent
```

Examples

- Calculate the ratio of requests whose status code is 200 to all requests

```
* | SELECT
 sum(
 CASE
 WHEN status =200 then 1
 ELSE 0 end
 ) *1.0 / count(1) as status_200_percentage
```

- Calculate the distribution of latencies

```
* | SELECT `
 CASE
 WHEN latency < 10 then 's10'
 WHEN latency < 100 then 's100'
 WHEN latency < 1000 then 's1000'
 WHEN latency < 10000 then 's10000'
 else 's_large' end
 as latency_slot,
 count(1) as pv
 group by latency_slot
```

Syntax for if() functions

An if() function works in the same way as a CASE statement does.

```
CASE
     WHEN condition THEN true_value
     [ ELSE false_value ]
 END
```

- if(condition, true_value)

  If the condition is true, the true_value column is returned. Otherwise, null is returned.

- if(condition, true_value, false_value)

  If the condition is true, the true_value column is returned. Otherwise, the false_value column is returned.

Syntax for coalesce() functions

A coalesce() function returns the first non-null value from multiple columns.

```
COALESCE (value1, value2 [,...])
```

Syntax for the nullif() function

If value1 is equal to value2, null is returned. Otherwise, value1 is returned.

```
nullif(value1, value2)
```

Syntax for the try() function

The try() function catches underlying exceptions, such as division by zero errors, and returns null.

```
try(expression)
```

28.1.4.8.20. Nested subqueries

This describes how to use nested subqueries when you query logs.

You can use nested queries to perform more complicated queries.

Nested queries differ from non-nested queries in the need for specifying the FROM clause in the SQL statement. You must specify the `from log` keyword in each SQL statement to read raw data from logs.

Example:

```
* | select sum(pv) from
(
select count(1) as pv from log group by method
)
```

28.1.4.8.21. Array functions

This topic describes the syntax of array functions. It also provides examples that show how to use these functions.

Function	Description	Example
Subscript operator []	The subscript operator [] is used to obtain an element in the array.	N/A
array_distinct	Returns the distinct elements in the array.	N/A
array_intersect(x, y)	Returns the intersection of the x and y arrays.	N/A
array_union(x, y) → array	Returns the union of the x and y arrays.	N/A
array_except(x, y) → array	Returns the subtraction of the x and y arrays.	N/A

Function	Description	Example
array_join(x, delimiter, null_replacement) → varchar	Joins string arrays with the delimiter into a string and replaces null values with null_replacement. **Note** The maximum size of the returned result of this array_join function is 1 KB. If the returned result exceeds 1 KB, the extra data will be truncated.	N/A
array_max(x) → x	Returns the maximum value in the x array.	N/A
array_min(x) → x	Returns the minimum value in the x array.	N/A
array_position(x, element) → bigint	Returns the subscript of the element in the x array. The subscript starts from 1. The value 0 is returned if no subscript is found.	N/A
array_remove(x, element) → array	Removes the element from the array.	N/A
array_sort(x) → array	Sorts the array and moves null values to the end.	N/A
cardinality(x) → bigint	Returns the array size.	N/A
concat(array1, array2, ..., arrayN) → array	Concatenates arrays.	N/A
contains(x, element) → boolean	Returns true if the x array contains the specified element.	N/A
filter(array, function) → array	For more information about this Lambda function, see the filter() function in Lambda functions.	N/A
flatten(x) → array	Flattens an array(array(T)) to an array(T) by concatenating the contained arrays.	N/A
reduce(array, initialState, inputFunction, outputFunction) → x	For more information, see the reduce() function in Lambda functions.	N/A
reverse(x) → array	Returns an array that has the reversed order of the x array.	N/A
sequence(start, stop) → array	Generates a sequence of elements from start to stop. The difference between elements is 1.	N/A
sequence(start, stop, step) → array	Generates a sequence of elements from start to stop. The difference between elements is step.	N/A

Function	Description	Example
sequence(start, stop, step) → array	Generates a sequence of timestamps from start to stop. The difference between timestamps is step. The type of step can be either INTERVAL DAY TO SECOND or INTERVAL YEAR TO MONTH.	N/A
shuffle(x) → array	Shuffles the array.	N/A
slice(x, start, length) → array	Returns a subset of the x array starting from the start value with the specified length.	N/A
transform(array, function) → array	For more information, see the transform() function in Lambda functions.	N/A
zip(array1, array2[, ...]) → array	Merges the specified arrays. The M-th element of the N-th argument will be the N-th field of the M-th output element.	`SELECT zip(ARRAY[1, 2], ARRAY['1b', null, '3b']); — [ROW(1, '1b'), ROW(2, null), ROW(null, '3b')]`
zip_with(array1, array2, function) → array	For more information, see the zip_with() function in Lambda functions.	N/A
array_agg (key)	An aggregate function that returns an array from values in the key column.	`* | select array_agg(key)`
array_transpose(array[array[*x*,*y*,*z*], array[*a*,*b*,*c*]])	Returns a new matrix by transposing the values of the previous matrix from rows to columns.	N/A

28.1.4.8.22. Binary string functions

This topic describes the syntax of binary string functions. It also provides examples that show how to use these functions.

Data of the VARBINARY type is different from data of the VARCHAR type.

Function	Description
Concatenation operator (‖)	The result of `a || b` is `ab`.
length(binary) → bigint	Returns the length in binary.
concat(binary1, ..., binaryN) → varbinary	Connects the binary strings, which is equivalent to ‖.
to_base64(binary) → varchar	Converts a binary string to a Base64 string.
from_base64(string) → varbinary	Converts a Base64 string to a binary string.
to_base64url(binary) → varchar	Converts a string to a URL-safe Base64 string.
from_base64url(string) → varbinary	Converts a URL-safe Base64 string to a binary string.

Function	Description
to_hex(binary) → varchar	Converts a binary string to a hexadecimal string.
from_hex(string) → varbinary	Converts a hexadecimal string to a binary string.
to_big_endian_64(bigint) → varbinary	Convert a number to a binary string in big endian mode.
from_big_endian_64(binary) → bigint	Converts a binary string in big endian mode to a number.
md5(binary) → varbinary	Calculates the MD5 value of a binary string.
sha1(binary) → varbinary	Calculates the SHA1 value of a binary string.
sha256(binary) → varbinary	Calculates the SHA256 hash value of a binary string.
sha512(binary) → varbinary	Calculate the SHA512 value of a binary string.
xxhash64(binary) → varbinary	Calculates the xxhash64 value of a binary string.

28.1.4.8.23. Bitwise operations

This topic describes the syntax for bitwise operations. It also provides examples that show how to use these operations.

Function	Description	Example

bit_count(x, bits) → bigint	Count the number of 1 in the binary expression of x.	`SELECT bit_count(9, 64); — 2`
		`SELECT bit_count(9, 8); — 2`
		`SELECT bit_count(-7, 64); — 62`
		`SELECT bit_count(-7, 8); — 6`

bitwise_and(x, y) → bigint	Perform the AND operation on x and y in the binary form.	N/A
bitwise_not(x) → bigint	Calculate the opposite values of all bits of x in the binary form.	N/A
bitwise_or(x, y) → bigint	Perform the OR operation on x and y in the binary form.	N/A
bitwise_xor(x, y) → bigint	Perform the XOR operation on x and y in the binary form.	N/A

28.1.4.8.24. Interval-valued comparison and periodicity-valued comparison functions

Log Service allows you to use interval-valued comparison and periodicity-valued comparison functions to query and analyze log data.

You can use the functions to compare the value for one period with that for a previous period.

Function	Description	Example
`compare(value, time_window)`	This function compares the value calculated for the current period with that calculated for the period before time_window. The data type of the values to be compared is Double or Long. The time_window parameter is measured in seconds. This function returns an array. Possible return values include the value for the current period, the value for the period before time_window, and the ratio of the current value to the value before time_window.	`*|select compare( pv , 86400) from (select count(1) as pv from log)`
`compare(value, time_window1, time_window2)`	This function compares the current value with the values for periods before time_window1 and time_window2. The comparison results are in the JSON array format, where the values must be returned in the following order: [current value, value before time_window1, value before time_window2, current value/value before time_window1, current value/value before time_window2].	`* | select compare(pv, 86400, 172800) from ( select count(1) as pv from log)`
`compare(value, time_window1, time_window2, time_window3)`	This function compares the value for the current period with the values for periods before time_window1, time_window2, and time_window3. The comparison results are in the JSON array format, where the values must be returned in the following order: [current value, value before time_window1, value before time_window2, value before time_window3, current value/value before time_window1, current value/value before time_window2, current value/value before time_window3].	`* | select compare(pv, 86400, 172800,604800) from ( select count(1) as pv from log)`

Function	Description	Example
`ts_compare(value, time_window)`	This function compares the value for the current period with the values for periods before `time_window1` and `time_window2` and returns a JSON array. The comparison results are in the JSON array format, where the values must be returned in the following order: [current value, value before time_window1, current value/value before time_window1, Unix timestamp that indicates the start time of the previous period]. This function is used to compare time series values. You must specify the GROUP BY keyword for the time column in SQL statements.	For example, `* | select t, ts_compare(pv, 86400 ) as d from(select date_trunc('minute',__time__ ) as t, count(1) as pv from log group by t order by t ) group by t` specifies that the function compares the calculation result of every minute in the current period with that of every minute in the last period. The comparison result is `d:[1251.0,1264.0, 0.9897151898734177, 1539843780.0,1539757380.0]t:2018-10-19 14:23:00.000` .

Examples

- Calculate the ratio of the PV for an hour on a day to that for the same time period on a previous day.

  The start time is 2018-07-25 14:00:00, and the end time is 2018-07-25 15:00:00.

  Statement for query and analysis:

  ```
  * | select compare( pv , 86400) from (select count(1) as pv from log)
  ```

  In the preceding statement, 86400 indicates 86,400 seconds before the current period.

  Return results:

  ```
  [9.0,19.0,0.47368421052631579]
  ```

  In the preceding results,

  - 9.0 is the PV for the period from 2018-07-25 14:00:00 to 2018-07-25 15:00:00.
  - 19.0 is the PV for the period from 2018-07-24 14:00:00 to 2018-07-24 15:00:00.
  - 0.47368421052631579 is the ratio of the PV for the current period to that for a previous period.

  If you want to expand the array into three columns of numbers, the analysis statement is

  ```
  * | select diff[1],diff[2],diff[3] from(select compare( pv , 86400) as diff from (select count(1) as pv from log))
  ```

- Calculate the PV ratio for every minute of the current hour to that in the same time period as the day before, and output the results in a line chart.

  i. Calculate the PV ratio for every minute of the current hour to that in the same time period as the day before. The start time is 2018-07-25 14:00:00, and the end time is 2018-07-25 15:00:00.

  Statement for query and analysis:

  ```
  *| select t, compare( pv , 86400) as diff from (select count(1) as pv, date_format(from_unixtime(__time__), '%H:%i') as t from log group by t) group by t order by t
  ```

  Return results:

t	diff
14:00	[9520.0,7606.0,1.2516434393899554]
14:01	[8596.0,8553.0,1.0050274757395066]
14:02	[8722.0,8435.0,1.0340248962655603]
14:03	[7499.0,5912.0,1.2684370771312586]

In the preceding table, t indicates the time in the format of `Hour:Minute`. The contents of the diff column are included in an array that contains the following values:

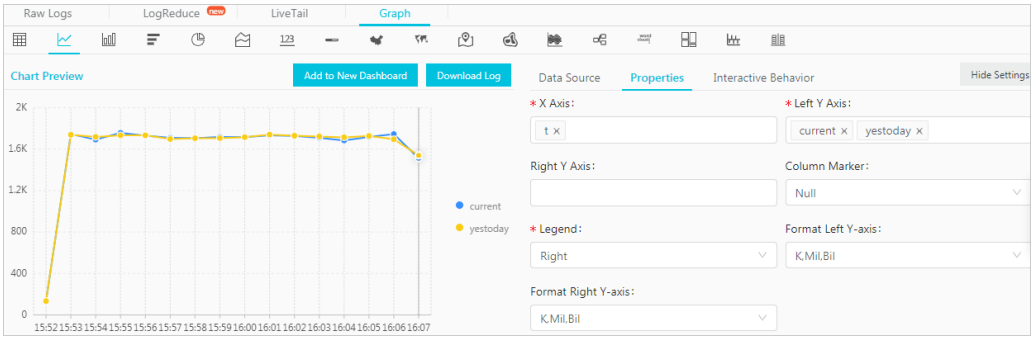
- The PV value of the current period.
- The PV value of the previous period.
- The ratio of the PV value for the current period to that for the previous period.

ii. To show the query results in a line chart, use the following statement:

```
*|select t, diff[1] as current, diff[2] as yestoday, diff[3] as percentage from(select t, compare( pv , 86400) as diff from (select count(1) as pv, date_format(from_unixtime(__time__), '%H:%i') as t from log group by t) group by t order by t)
```

The two lines indicate the PV values of a day and the day before.

Line chart



28.1.4.8.25. Comparison functions and operators

This topic describes the comparison functions and operators in Log Service. You can use these functions and operators to query and analyze log data.

Comparison functions and operators

A comparison function compares two parameter values of any comparable data types, such as INTEGER, BIGINT, DOUBLE, and TEXT.

Comparison operators

A comparison operator is used to compare two values. If the statement is true, TRUE is returned. Otherwise, FALSE is returned.

Operator	Description
<	Less than
>	Greater than

Operator	Description
<=	Less than or equal to
>=	Greater than or equal to
=	Equal to
<>	Not equal to
! =	Not equal to

Range operator BETWEEN

The BETWEEN operator determines whether a value falls in a specified closed interval.

- If the value falls in the specified closed interval, TRUE is returned. Otherwise, FALSE is returned.

  Example: `SELECT 3 BETWEEN 2 AND 6;` . The statement is true, and TRUE is returned.

  The preceding statement is equivalent to `SELECT 3 >= 2 AND 3 <= 6;` .
- The BETWEEN operator can be put behind the NOT operator to test whether a value falls out of a specified closed interval.

  Example: `SELECT 3 NOT BETWEEN 2 AND 6;` . The statement is false, and FALSE is returned.

  The preceding statement is equivalent to `SELECT 3 < 2 OR 3 > 6;` .
- If any of the three values is NULL, NULL is returned.

IS NULL and IS NOT NULL

The IS NULL and IS NOT NULL operators test whether a value is NULL.

IS DISTINCT FROM and IS NOT DISTINCT FROM

These operators are similar to the EQUAL TO and NOT EQUAL TO operators. However, IS DISTINCT FROM and IS NOT DISTINCT FROM can determine whether a NULL value exists.

Examples:

```
SELECT NULL IS DISTINCT FROM NULL; -- false
SELECT NULL IS NOT DISTINCT FROM NULL; -- true
```

The DISTINCT operator compares parameter values under multiple conditions, as described in the following table.

a	b	a = b	a <> b	a DISTINCT b	a NOT DISTINCT b
1	1	TRUE	FALSE	FALSE	TRUE
1	2	FALSE	TRUE	TRUE	FALSE
1	NULL	NULL	NULL	TRUE	FALSE
NULL	NULL	NULL	NULL	FALSE	TRUE

GREATEST and LEAST

These operators are used to obtain the maximum or minimum value from a row of field values.

Example:

```
select greatest(1,2,3) -- Returns 3.
```

Quantified comparison predicates: ALL, ANY, and SOME

The ALL, ANY, and SOME quantifiers can be used to determine whether a parameter value meets specified conditions.

- ALL is used to determine whether a parameter value meets all conditions. If the statement is true, TRUE is returned. Otherwise, FALSE is returned.
- ANY is used to determine whether a parameter value meets a condition. If the statement is true, TRUE is returned. Otherwise, FALSE is returned.
- SOME is used to determine whether a parameter value meets a condition. SOME is equivalent to ANY.
- ALL, ANY, and SOME must immediately follow comparison operators.

ALL and ANY support comparison under multiple conditions, as described in the following table.

Expression	Description
A = ALL (...)	Returns TRUE if A matches all values.
A <> ALL (...)	Returns TRUE if A does not match a value.
A < ALL (...)	Returns TRUE if A is smaller than the smallest value.
A = ANY (...)	Returns TRUE if A is equal to a value. This statement is equivalent to A IN (...).
A <> ANY (...)	Returns TRUE if A does not match a value.
A < ANY (...)	Returns TRUE if A is smaller than the largest value.

Examples:

```
SELECT 'hello' = ANY (VALUES 'hello', 'world'); -- true
SELECT 21 < ALL (VALUES 19, 20, 21); -- false
SELECT 42 >= SOME (SELECT 41 UNION ALL SELECT 42 UNION ALL SELECT 43); -- true
```

28.1.4.8.26. Lambda functions

This topic describes Lambda functions and provides some examples. You can use Lambda functions to analyze log data in Log Service

Lambda expressions

Lambda expressions use the arrow operator `->`.

Examples:

```
x -> x + 1
(x, y) -> x + y
x -> regexp_like(x, 'a+')
x -> x[1] / x[2]
x -> IF(x > 0, x, -x)
x -> COALESCE(x, 0)
x -> CAST(x AS JSON)
x -> x + TRY(1 / 0)
```

Most MySQL expressions can be used in Lambda functions.

filter(array<T>, function<T, boolean>) → ARRAY<T>

Returns an array whose elements are filtered from the specified array based on the Lambda expression.

Examples:

```
SELECT filter(ARRAY [], x -> true); -- []
SELECT filter(ARRAY [5, -6, NULL, 7], x -> x > 0); -- [5, 7]
SELECT filter(ARRAY [5, NULL, 7, NULL], x -> x IS NOT NULL); -- [5, 7]
```

map_filter(map<K, V>, function<K, V, boolean>) → MAP<K,V>

Returns a map whose elements are filtered based on the Lambda expression. The map is generated from the map function.

Examples:

```
SELECT map_filter(MAP(ARRAY[], ARRAY[]), (k, v) -> true); -- {}
SELECT map_filter(MAP(ARRAY[10, 20, 30], ARRAY['a', NULL, 'c']), (k, v) -> v IS NOT NULL); -- {10 ->
a, 30 -> c}
SELECT map_filter(MAP(ARRAY['k1', 'k2', 'k3'], ARRAY[20, 3, 15]), (k, v) -> v > 10); -- {k1 -> 20, k3
-> 15}
```

reduce(array<T>, initialState S, inputFunction<S, T, S>, outputFunction<S, R>) → R

The reduce function starts from the initial state, traverses each element in the array, and then calls inputFunction(S,T) to generate a new state. After all the elements in the array are traversed and the final state is generated, the reduce function calls outputFunction to assign the final state value to the result R and output the result. The procedure is described as follows:

1. Start from the initial state S.
2. Traverse each element T.
3. Calculate inputFunction(S,T) to generate a new state S.
4. Repeat steps 2 and 3 until the last element is traversed and has a new state.
5. Turn the final state S into the final result R.

Examples:

```
SELECT reduce(ARRAY [], 0, (s, x) -> s + x, s -> s); -- 0
SELECT reduce(ARRAY [5, 20, 50], 0, (s, x) -> s + x, s -> s); -- 75
SELECT reduce(ARRAY [5, 20, NULL, 50], 0, (s, x) -> s + x, s -> s); -- NULL
SELECT reduce(ARRAY [5, 20, NULL, 50], 0, (s, x) -> s + COALESCE(x, 0), s -> s); -- 75
SELECT reduce(ARRAY [5, 20, NULL, 50], 0, (s, x) -> IF(x IS NULL, s, s + x), s -> s); -- 75
SELECT reduce(ARRAY [2147483647, 1], CAST (0 AS BIGINT), (s, x) -> s + x, s -> s); -- 2147483648
SELECT reduce(ARRAY [5, 6, 10, 20], -- calculates arithmetic average: 10.25
              CAST(ROW(0.0, 0) AS ROW(sum DOUBLE, count INTEGER)),
              (s, x) -> CAST(ROW(x + s.sum, s.count + 1) AS ROW(sum DOUBLE, count INTEGER)),
              s -> IF(s.count = 0, NULL, s.sum / s.count));
```

transform(array<T>, function<T, U>) → ARRAY<U>

This Lambda function traverses each element in an array to generate a new result U.

Examples:

```
SELECT transform(ARRAY [], x -> x + 1); -- []
SELECT transform(ARRAY [5, 6], x -> x + 1); -- [6, 7] -- Increments each element by 1.
SELECT transform(ARRAY [5, NULL, 6], x -> COALESCE(x, 0) + 1); -- [6, 1, 7]
SELECT transform(ARRAY ['x', 'abc', 'z'], x -> x || '0'); -- ['x0', 'abc0', 'z0']
SELECT transform(ARRAY [ARRAY [1, NULL, 2], ARRAY[3, NULL]], a -> filter(a, x -> x IS NOT NULL)); --
[[1, 2], [3]]
```

zip_with(array<T>, array<U>, function<T, U, R>) → array<R>

This Lambda function merges two arrays and generates the element R in the new array based on element T and element U.

Examples:

```
SELECT zip_with(ARRAY[1, 3, 5], ARRAY['a', 'b', 'c'], (x, y) -> (y, x)) --Transposes the elements of
the two arrays to generate a new array. Result: [['a', 1], ['b', 3],['c', 5]]
SELECT zip_with(ARRAY[1, 2], ARRAY[3, 4], (x, y) -> x + y); -- Result: [4, 6]
SELECT zip_with(ARRAY['a', 'b', 'c'], ARRAY['d', 'e', 'f'], (x, y) -> concat(x, y)) -- Concatenates t
he elements of the two arrays to generate a new string. Result: ['ad', 'be', 'cf']
```

28.1.4.8.27. Logical functions

This topic describes the available logical functions in Log Service. You can use these functions to query and analyze log data.

Logical operators

Operator	Description	Example
AND	The result is TRUE if both values are TRUE.	a AND b
OR	The result is TRUE if either value is TRUE.	a OR b
NOT	The result is TRUE if the value is FALSE.	NOT a

Effect of NULL on logical operators

The following tables list the truth values when the values of a and b are TRUE, FALSE, and NULL, respectively.

Truth table 1

a	b	a AND b	a OR b
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
TRUE	NULL	NULL	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE
FALSE	NULL	FALSE	NULL

a	b	a AND b	a OR b
NULL	TRUE	NULL	TRUE
NULL	FALSE	FALSE	NULL
NULL	NULL	NULL	NULL

Truth table 2

a	NOT a
TRUE	FALSE
FALSE	TRUE
NULL	NULL

28.1.4.8.28. Field aliases

This topic describes how to specify an alias for a field and provides some examples.

A field name in an SQL statement must start with letters and contain digits and underscores (_).

If you have configured a field name that does not conform to the SQL standard (such as User-Agent), you must specify an alias for the field on the field index configuration page. The alias takes effect only for the duration of the SQL statement. The data is still stored under the original field name. You must specify the original name when you perform a search.

You can also specify an alias for a field in an SQL statement if the original name is long.

Sample aliases

Original field name	Alias
User-Agent	ua
User.Agent	ua
123	col
abceefghijklmnopqrstuvw	a

28.1.4.8.29. JOIN operations between Logstores and Relational Database Service (RDS) tables

This topic describes how to join Logstores in Log Service with RDS tables for queries and store the query results in RDS tables.

Procedure

1. Create a VPC.

   Create an RDS instance and specify the VPC to host the RDS instance. Then the VPC ID and the RDS instance ID are obtained.

2. Configure a whitelist for the RDS instance.

Add the following CIDR blocks to the whitelist: `100.104.0.0/16`, `11.194.0.0/16`, and `11.201.0.0/16`

3. Create an external store

   Run the following statement to create an external store. Replace the parameter values based on your business needs.

```
{
"externalStoreName":"storeName",
"storeType":"rds-vpc",
"parameter":
   {
   "region":"cn-qingdao",
   "vpc-id":"vpc-m5eq4irc1pucp*******"
   "instance-id":"i-m5eeo2whsn*******"
   "host":"localhost",
   "port":"3306",
   "username":"root",
   "password":"****",
   "db":"scmc"
   "table":"join_meta"
   }
}
```

   Parameters

Parameter	Description
region	The region where your RDS instance resides.
vpc-id	The ID of the VPC where your RDS instance resides.
instance-id	The ID of the RDS instance.
host	The ID of the ECS instance that is used to access the RDS instance.
port	The port of the ECS instance that is used to access the RDS instance.
username	The username that is used to log on to the RDS instance.
password	The password that is used to log on to the RDS instance.
db	The name of the database.
table	The name of the table with which the Logstore is joined.

   Note You can join a Logstore with an RDS table that resides only in the China (Beijing), China (Qingdao), and China (Hangzhou) regions.

4. JOIN query.

   Log on to the Log Service console. In the **Search & Analyze** search box, run a JOIN statement.

   Supported JOIN syntax:

   - INNER JOIN

- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

```
[ INNER ] JOIN
LEFT [ OUTER ] JOIN
RIGHT [ OUTER ] JOIN
FULL [ OUTER ] JOIN
```

> **Note**
>
> - You can join Logstores only to external tables.
> - In the JOIN statement, you must first specify a Logstore before specifying an external store.
> - You must specify the name of the external store instead of the name of an RDS table. The external store name automatically changes into the combination of the RDS database name and the name of the RDS table that you want to join with the Logstore.

Sample JOIN statement:

```
method:postlogstorelogs | select count(1) , histogram(logstore) from log  l join join_meta m o
n l.projectid = cast( m.ikey as varchar)
```

5. Store the query results to the RDS table.

   You can use the INSERT statement to insert the query results into the RDS table.

```
method:postlogstorelogs | insert into method_output  select cast(methodasvarchar(65535)),count
(1)fromloggroupbymethod
```

Sample Python script:

```
# encoding: utf-8
from __future__ import print_function
from aliyun.log import *
from aliyun.log.util import base64_encodestring
from random import randint
import time
import os
from datetime import datetime
    endpoint = os.environ.get('ALIYUN_LOG_SAMPLE_ENDPOINT', 'cn-chengdu.log.aliyuncs.com')
    accessKeyId = os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSID', '')
    accessKey = os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSKEY', '')
    logstore = os.environ.get('ALIYUN_LOG_SAMPLE_LOGSTORE', '')
    project = "ali-yunlei-chengdu"
    client = LogClient(endpoint, accessKeyId, accessKey, token)
## Create an external store
    res = client.create_external_store(project,ExternalStoreConfig("rds_store","region","rds-vpc","vpc id","instance-id","instance-ip","port","username","password","db","table"));
    res.log_print()
    ## Obtain external store details
    res = client.get_external_store(project,"rds_store");
    res.log_print()
    res = client.list_external_store(project,"");
    res.log_print();
    # Perform the JOIN operation.
    req = GetLogStoreLogsRequest(project,logstore,From,To,"","select count(1) from  "+ logstore +"  s join  meta m on  s.projectid = cast(m.ikey as varchar)");
    res = client.get_logs(req)
    res.log_print();
     # Store query results to the RDS table
    req = GetLogStoreLogsRequest(project,logstore,From,To,"," insert into rds_store select count(1) from  "+ logstore );
    res = client.get_logs(req)
    res.log_print();
```

28.1.4.8.30. Geospatial functions

This topic describes the available geospatial functions in Log Service. You can use these functions to query and analyze log data.

Concept of geometry

Geospatial functions support geometries in the well-known text (WKT) format.

Geometry formats

Geometry	WKT format
Point	`POINT (0 0)`
LineString	`LINESTRING (0 0, 1 1, 1 2)`
Polygon	`POLYGON ((0 0, 4 0, 4 4, 0 4, 0 0), (1 1, 2 1, 2 2, 1 2, 1 1))`
MultiPoint	`MULTIPOINT (0 0, 1 2)`

Geometry	WKT format
MultiLineString	`MULTILINESTRING ((0 0, 1 1, 1 2), (2 3, 3 2, 5 4))`
MultiPolygon	`MULTIPOLYGON (((0 0, 4 0, 4 4, 0 4, 0 0), (1 1, 2 1, 2 2, 1 2, 1 1)), ((-1 -1, -1 -2,`
GeometryCollection	`GEOMETRYCOLLECTION (POINT(2 3), LINESTRING (2 3, 3 4))`

Constructors

Constructor description

Function	Description
ST_Point(double, double) → Point	Returns a geometry point instance with the specified coordinate values.
ST_LineFromText(varchar) → LineString	Returns a geometry LineString instance from a WKT representation.
ST_Polygon(varchar) → Polygon	Returns a geometry polygon instance from a WKT representation.
ST_GeometryFromText(varchar) → Geometry	Returns a geometry instance from a WKT representation.
ST_AsText(Geometry) → varchar	Returns the WKT representation of a geometry.

Operations

Function	Description
ST_Boundary(Geometry) → Geometry	Returns the closure of the combinatorial boundary of a geometry.
ST_Buffer(Geometry, distance) → Geometry	Returns the geometry that represents all points whose distance from the specified geometry is shorter than or equal to the specified distance.
ST_Difference(Geometry, Geometry) → Geometry	Returns the geometry value that represents the point set difference of the specified geometries.
ST_Envelope(Geometry) → Geometry	Returns the bounding rectangular polygon of a geometry.
ST_ExteriorRing(Geometry) → Geometry	Returns a line string that represents the exterior ring of the input polygon.
ST_Intersection(Geometry, Geometry) → Geometry	Returns the geometry value that represents the point set intersection of two geometries.
ST_SymDifference(Geometry, Geometry) → Geometry	Returns the geometry value that represents the point set symmetric difference of two geometries.

Relationship tests

Function	Description
ST_Contains(Geometry, Geometry) → boolean	Returns True if and only if no points of the second geometry lie in the exterior of the first geometry, and at least one point of the interior of the first geometry lies in the interior of the second geometry. Returns False if points of the second geometry are on the boundary of the first geometry.
ST_Crosses(Geometry, Geometry) → boolean	Returns True if the specified geometries share some, but not all, interior points in common.
ST_Disjoint(Geometry, Geometry) → boolean	Returns True if the specified geometries do not spatially intersect.
ST_Equals(Geometry, Geometry) → boolean	Returns True if the specified geometries represent the same geometry.
ST_Intersects(Geometry, Geometry) → boolean	Returns True if the specified geometries spatially intersect in two dimensions.
ST_Overlaps(Geometry, Geometry) → boolean	Returns True if the specified geometries share space in the same dimension, but are not completely contained by each other.
ST_Relate(Geometry, Geometry) → boolean	Returns True if the first geometry is spatially related to the second geometry.
ST_Touches(Geometry, Geometry) → boolean	Returns True if the specified geometries have at least one point in common, but their interiors do not intersect.
ST_Within(Geometry, Geometry) → boolean	Returns True if the first geometry is completely inside the second geometry. Returns False if the two geometries have points in common at the boundaries.

Accessors

Function	Description
ST_Area(Geometry) → double	Returns the two-dimensional Euclidean area of a geometry.
ST_Centroid(Geometry) → Geometry	Returns the point value that is the mathematical centroid of a geometry.
ST_CoordDim(Geometry) → bigint	Returns the coordinate dimension of a geometry.
ST_Dimension(Geometry) → bigint	Returns the inherent dimension of a geometry object, which must be less than or equal to the coordinate dimension.
ST_Distance(Geometry, Geometry) → double	Returns the minimum two-dimensional Cartesian distance (based on spatial ref) between two geometries in projected units.
ST_IsClosed(Geometry) → boolean	Returns True if the start and end points of the linestring are coincident.

Function	Description
ST_IsEmpty(Geometry) → boolean	Returns True if the specified geometry is an empty geometry, such as geometry collection, polygon, and point.
ST_IsRing(Geometry) → boolean	Returns True if and only if the line is closed and simple.
ST_Length(Geometry) → double	Returns the length of a LineString or multi-LineString by using Euclidean measurement on a two-dimensional plane (based on spatial ref) in projected units.
ST_XMax(Geometry) → double	Returns the X maximum of the bounding box of the geometry.
ST_YMax(Geometry) → double	Returns the Y maximum of the bounding box of the geometry.
T_XMin(Geometry) → double	Returns the X minimum of the bounding box of the geometry.
ST_YMin(Geometry) → double	Returns the Y minimum of the bounding box of the geometry.
ST_StartPoint(Geometry) → point	Returns the first point of a geometry LineString instance.
ST_EndPoint(Geometry) → point	Returns the last point of a geometry LineString instance.
ST_X(Point) → double	Returns the X coordinate of a point.
ST_Y(Point) → double	Returns the Y coordinate of a point.
ST_NumPoints(Geometry) → bigint	Returns the number of points in a geometry.
ST_NumInteriorRing(Geometry) → bigint	Returns the cardinality of the collection of interior rings of a polygon.

28.1.4.8.31. Geography functions

This topic describes the syntax of geography functions and provides some examples.

For information about functions that identify the country, province, city, ISP, and the longitude and latitude of a specified IP address, see IP functions.

Geography functions

Function	Description	Example
geohash(string)	Returns the geohash value of the specified geographical location. The geographical location is represented by a string in the format of "latitude, longitude". The values for latitude and longitude are separated by a comma.	`select geohash('34.1,120.6')= 'wwjcbrdnzs'`

Function	Description	Example
geohash(lat,lon)	Returns the geohash value of the specified geographical location. The geographical location is represented by two parameters that indicate the latitude and longitude.	`select geohash(34.1,120.6)= 'wwjcbrdnzs'`

28.1.4.8.32. JOIN syntax

The JOIN operation joins multiple tables by using one or more fields in the tables. You can join a Logstore created in Log Service with the Logstore itself, with another Logstore, or with an RDS table. This topic describes how to join different Logstores.

Procedure

1. Download the latest version of the SDK for Python.
2. Call the GetProjectLogs operation to query logs.

Sample SDK

```
#! /usr/bin/env python
#encoding: utf-8
import time,sys,os
from aliyun.log.logexception import LogException
from aliyun.log.logitem import LogItem
from aliyun.log.logclient import LogClient
from aliyun.log.getlogsrequest import GetLogsRequest
from aliyun.log.getlogsrequest import GetProjectLogsRequest
from aliyun.log.putlogsrequest import PutLogsRequest
from aliyun.log.listtopicsrequest import ListTopicsRequest
from aliyun.log.listlogstoresrequest import ListLogstoresRequest
from aliyun.log.gethistogramsrequest import GetHistogramsRequest
from aliyun.log.index_config import *
from aliyun.log.logtail_config_detail import *
from aliyun.log.machine_group_detail import *
from aliyun.log.acl_config import *
if __name__=='__main__':
    token = None
    endpoint = "http://cn-hangzhou.log.aliyuncs.com"
    accessKeyId  = 'LTAIvKy7U'
    accessKey='6gXLNTLyCfdsfwrewrfhdskfdsfuiwu'
    client = LogClient(endpoint, accessKeyId, accessKey,token)
    logstore = "meta"
    # In the query statements, specify two Logstores, the query time ranges of both Logstores, and th
e key that you want to use to join the two Logstores.
    req = GetProjectLogsRequest(project,"select count(1) from sls_operation_log  s join  meta m on s.
__date__ >'2018-04-10 00:00:00' and s.__date__ < '2018-04-11 00:00:00' and m.__date__ >'2018-04-23 00
:00:00' and m.__date__ <'2018-04-24 00:00:00' and s.projectid = cast(m.ikey as varchar)");
    res = client.get_project_logs(req)
    res.log_print();
    exit(0)
```

Note For more information about the JOIN syntax and examples, see Join.

28.1.4.8.33. UNNEST function

This topic describes the UNNEST function.

Scenarios

Log data is typically stored as primitive data types, such as string or number. In certain scenarios, log data may include complex data types, such as arrays, maps, and JSON objects. The UNNEST function can be used to transform complex data types into rows of primitive data types. This simplifies query and analysis.

Example:

```
__source__:  1.1.1.1
__tag__:__hostname__:  vm-req-170103232316569850-tianchi111932.tc
__topic__:  TestTopic_4
array_column:  [1,2,3]
double_column:  1.23
map_column:  {"a":1,"b":2}
text_column: Product
```

The values of the `array_column` field are arrays. To obtain the sum of elements of all `array_column` field values, you must traverse all elements of every array.

UNNEST function

Syntax	Description
`unnest( array) as table_alias(column_name)`	Expands an array into multiple rows. The column name of these rows is `column_name`.
`unnest(map) as table(key_name, value_name)`	Expands a map into multiple rows. `key_name` specifies the column name of the keys, and `value_name` specifies the column name of the values.

Note The UNNEST function is used to expand arrays or maps. If you want to expand a string, you must transform the string into a JSON object, and then convert the JSON object into an array or map. To do this, you can use the `cast(json_parse(array_column) as array(bigint))` function.

Traverse every element of an array

Expands an array into multiple rows by using the following SQL SELECT statement:

```
* | select  array_column, a  from log, unnest( cast( json_parse(array_column)  as array(bigint) ) )
as  t(a)
```

The UNNEST function `unnest( cast( json_parse(array_column) as array(bigint) ) ) as t(a)` expands the array into multiple rows. The rows are stored in a derived table referenced as t, with the column referenced as a.

- Calculate the sum of the elements in an array:

  ```
  * | select  sum(a)  from log, unnest( cast( json_parse(array_column)  as array(bigint) ) ) as
  t(a)
  ```

- Perform a GROUP BY operation on all elements of an array:

  ```
  * | select  a, count(1)  from log, unnest( cast( json_parse(array_column)  as array(bigint) )
  ) as  t(a)  group by a
  ```

Traverse every key and value of a map

- Traverse every key and value of a map:

```
* | select  map_column , a,b    from log, unnest( cast( json_parse(map_column)   as map(varchar, bigint) ) ) as  t(a,b)
```

- Perform a GROUP BY operation on all keys of a map:

```
* | select   key,  sum(value)    from log, unnest( cast( json_parse(map_column)   as map(varchar, bigint) ) ) as  t(key,value)    GROUP  BY  key
```

Visualize the query results of the histogram and numeric_histogram functions.

- histogram

  The histogram function works in a similar manner to the count group by syntax. For more information about the histogram function, see Map functions.

  In most cases, the histogram function returns a JSON object. The following is an example:

```
* | select histogram(method)
```

  You can use the UNNEST function to expand JSON data into multiple rows. Then the data can be visualized. The following is an example:

```
* | select  key , value  from( select histogram(method) as his from log) , unnest(his ) as t(key,value)
```

- numeric_histogram

  The numeric_histogram function assigns a column of numeric values into multiple bins. This function is equivalent to a GROUP BY operation that is performed on a numeric value column. For more information about the syntax of the numeric_histogram function, see Approximate functions.

```
* | select numeric_histogram(10,Latency)
```

  Use the following SELECT statement to visualize the result:

```
* |  select key,value from(select numeric_histogram(10,Latency) as his from log) , unnest(his) as t(key,value)
```

28.1.4.9. Machine learning syntax and functions

28.1.4.9.1. Overview

The machine learning feature of Log Service supports multiple algorithms and calling methods. You can use SELECT statements and machine learning functions to analyze the characteristics of a field or fields within a period of time.

Log Service offers multiple time series analysis algorithms to help you implement time series prediction, time series anomaly detection, time series decomposition, and multi-time series clustering. The algorithms are compatible with standard SQL statements. This greatly simplifies the use of the algorithms and improves the troubleshooting efficiency.

Features

- Supports various smooth operations on single-time series data.
- Supports algorithms related to the prediction, anomaly detection, change point detection, inflection point

detection, and multi-period estimation of single-time series data.

- Supports decomposition operations on single-time series data.
- Supports various clustering algorithms of multi-time series data.
- Supports multi-field pattern mining (based on the sequence of numeric data or text).

Limits

- The specified time series data must be sampled based on the same interval.
- The specified time series data cannot contain data repeatedly sampled from the same time point.

Item	Description
Processing capacity of time-series data	Data can be collected from a maximum of 150,000 consecutive time points. If the data volume exceeds the processing capacity, you must aggregate the data or reduce the sampling amount.
Clustering capacity of the density-based clustering algorithm	A maximum of 5,000 time series curves, each of which cannot contain more than 1,440 time points.
Clustering capacity of the hierarchical clustering algorithm	A maximum of 2,000 time series curves, each of which cannot contain more than 1,440 time points.

Machine learning functions

Type	Function	Description
Smooth functions	ts_smooth_simple	Uses the Holt Winters algorithm to smooth time series data.
	ts_smooth_fir	Uses the finite impulse response (FIR) filter to smooth time series data.
	ts_smooth_iir	Uses the infinite impulse response (IIR) filter to smooth time series data.
Multi-period estimation functions	ts_period_detect	Forecasts time series data by period.
Change point detection functions	ts_cp_detect	Finds intervals with different statistical characteristics from time series data. The interval endpoints are change points.
	ts_breakout_detect	Finds the time points when statistics steeply increases or decreases from time series data.
Maximum value detection function	ts_find_peaks	Finds the local maximum value of time series data in a specified window.
	ts_predicate_simple	Uses default parameters to model time series data and performs simple time series prediction and anomaly detection.
	ts_predicate_ar	Uses an autoregressive (AR) model to model time series data and performs simple time series prediction and anomaly detection.

Type	Function	Description
Prediction and anomaly detection functions	ts_predicate_arma	Uses an autoregressive moving average (ARMA) model to model time series data and performs simple time series prediction and anomaly detection.
	ts_predicate_arima	Uses an autoregressive integrated moving average (ARIMA) model to model time series data and performs simple time series prediction and anomaly detection.
	ts_regression_predict	Accurately predicts the trend for a single periodic time series with a certain tendency.
Time series decomposition function	ts_decompose	Uses the Seasonal and Trend decomposition using Loess (STL) algorithm to decompose time series data.
Time series clustering functions	ts_density_cluster	Uses a density-based clustering method to cluster multiple pieces of time series data.
	ts_hierarchical_cluster	Uses a hierarchical clustering method to cluster multiple pieces of time series data.
	ts_similar_instance	Queries curves that are similar to a specified curve.
Frequent pattern statistics function	pattern_stat	Mines representative combinations of attributes among the given multi-attribute field samples to obtain the frequent pattern in statistical patterns.
Differential pattern statistics function	pattern_diff	Finds the pattern that causes differences between two collections under specified conditions.
Root cause analysis function	rca_kpi_search	When a time series metric is abnormal, you can use the root cause analysis function to analyze the dimension attributes that result in the abnormal metric in a timely manner.
Correlation analysis functions	ts_association_analysis	Quickly finds the metrics that are correlated with a specified metric among multiple observed metrics in the system.
ts_similar	Quickly finds the metrics that are correlated with specified time series data among multiple observed metrics in the system.	
Kernel density estimation function	kernel_density_estimation	Uses the smooth peak function to fit the observed data points, thus simulating the real probability distribution curve.

28.1.4.9.2. Smooth functions

This topic describes the smooth functions that you can use to smooth and filter specified time series curves. Filtering is the first step to discover the shapes of time series curves.

Functions

Function	Description
ts_smooth_simple	Uses the Holt-Winters algorithm to filter time series data. This function is the default smooth function.
ts_smooth_fir	Uses a finite impulse response (FIR) filter to filter time series data.
ts_smooth_iir	Uses an infinite impulse response (IIR) filter to filter time series data.

ts_smooth_simple

- Syntax:

```
select ts_smooth_simple(x, y)
```

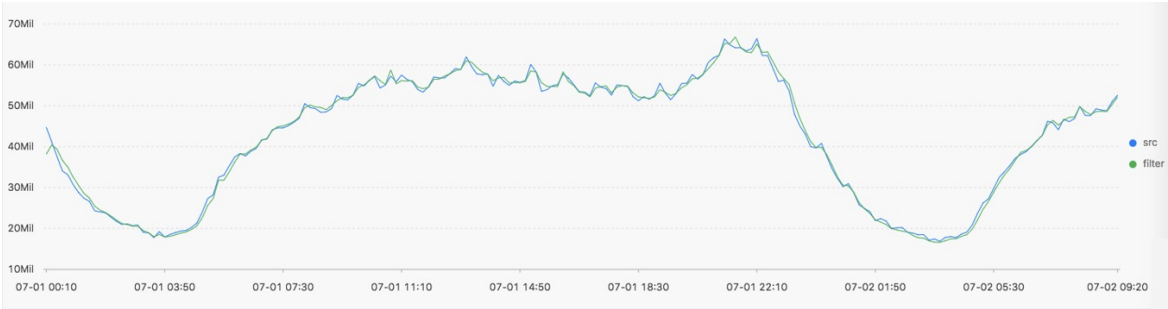
- The following table lists the parameters of the function.

Parameter	Description	Value
x	The time sequence. The time points along the x axis are sorted in the ascending order.	The Unix timestamp of the time series data. Unit: seconds.
y	The sequence of numeric data at each specified time point.	-

- Example
  - The search and analytic statement is shown as follows:

```
* | select ts_smooth_simple(stamp, value) from ( select __time__ - __time__ % 120 as stamp, avg(
v) as value from log GROUP BY stamp order by stamp )
```

  - The following figure shows the response.



- The following table lists the display items.

Item		Description
Horizontal axis	unixtime	The Unix timestamp of time series data. Unit: seconds.
Vertical axis	src	The unfiltered data.
	filter	The filtered data.

ts_smooth_fir

- Syntax:

- If you cannot determine filter parameters, use built-in window parameters in the following statement:

```
select ts_smooth_fir(x, y,winType,winSize)
```

- If you can determine filter parameters, you can specify the parameters as needed in the following statement:

```
select ts_smooth_fir(x, y,array[])
```

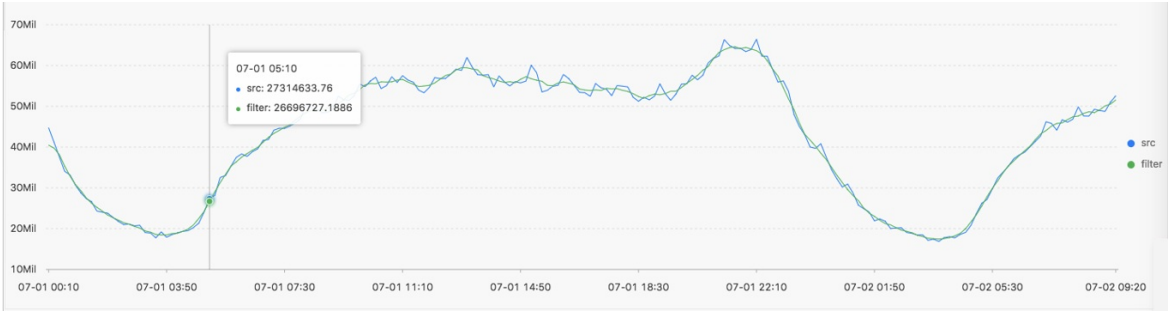
- The following table lists the parameters of the function.

Parameter	Description	Value
x	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
y	The sequence of numeric data at each specified time point.	-
winType	The type of window used for filtering.	Valid values: ○ rectangle: rectangle window. ○ hanning: hanning window ○ hamming: hamming window. ○ blackman: blackman window. **Note** We recommend that you select the rectangle window for better display effects.
winSize	The length of the filtering window.	The value is of the LONG type. Valid values: 2 to 15.
array[]	The parameter used for FIR filtering.	The value is an array where the sum of elements is 1. For example, array[0.2, 0.4, 0.3, 0.1].

- Example 1
  - The search and analytic statement is shown as follows:

```
* | select ts_smooth_fir(stamp, value, 'rectangle', 4) from ( select __time__ - __time__ % 120 a
s stamp, avg(v) as value from log GROUP BY stamp order by stamp )
```

  - The following figure shows the response.

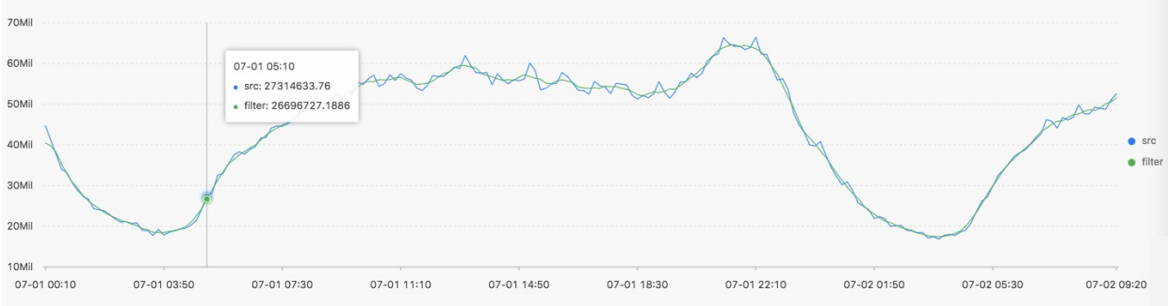


- Example 2

- The search and analytic statement is shown as follows:

```
* | select ts_smooth_fir(stamp, value, array[0.2, 0.4, 0.3, 0.1]) from ( select __time__ - __time__ % 120 as stamp, avg(v) as value from log GROUP BY stamp order by stamp )
```

- The following figure shows the response.



- The following table lists the display items.

Item		Description
Horizontal axis	unixtime	The Unix timestamp of the time series data. Unit: seconds.
Vertical axis	src	The unfiltered data.
	filter	The filtered data.

ts_smooth_iir

- Syntax:

```
select ts_smooth_iir(x, y, array[], array[] )
```

- The following table lists the parameters of the function.

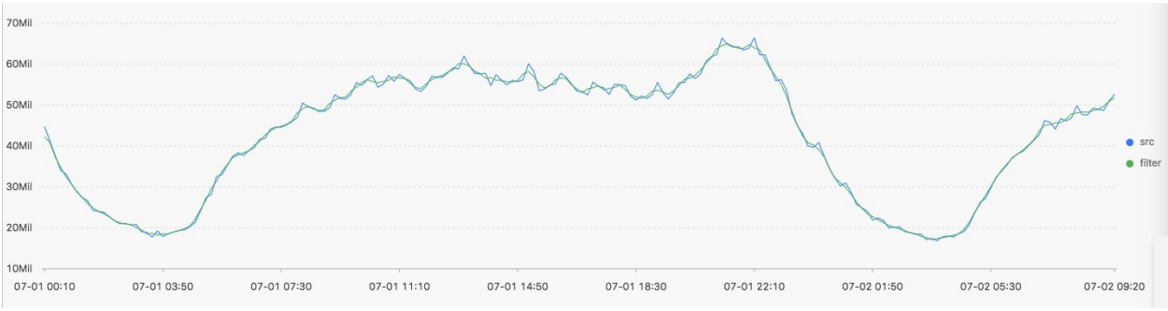
Parameter	Description	Value
x	The time sequence. The time points along the x axis are sorted in ascending order.	Each time point is a Unix timestamp. Unit: seconds.
y	The sequence of numeric data at each specified time point.	-
array[]	The parameter used for IIR filtering in terms of x_i.	The value is an array where the sum of elements is 1. The length of the array ranges from 2 to 15. For example, array[0.2, 0.4, 0.3, 0.1].
array[]	The parameter used for IIR filtering in terms of y_{i-1}.	The value is an array where the sum of elements is 1. The length of the array ranges from 2 to 15. For example, array[0.2, 0.4, 0.3, 0.1].

- Example

- The search and analytic statement is shown as follows:

```
* | select ts_smooth_iir(stamp, value, array[0.2, 0.4, 0.3, 0.1], array[0.4, 0.3, 0.3]) from ( select __time__ - __time__ % 120 as stamp, avg(v) as value from log GROUP BY stamp order by stamp )
```

- The following figure shows the response.



- The following table lists the display items.

Item		Description
Horizontal axis	unixtime	The Unix timestamp of the time series data. Unit: seconds.
Vertical axis	src	The unfiltered data.
	filter	The filtered data.

28.1.4.9.3. Multi-period estimation functions

This topic describes the multi-period estimation functions that are supported by Log Service. You can use the functions to estimate the periods of time series data in different time intervals. You can also extract the periods by performing a series of operations such as Fourier transform (FT).

Functions

Function	Description
ts_period_detect	Estimates the periods of time series data that is distributed in different time intervals.
ts_period_classify	Uses FT to calculate the periodicity of specified time series curves. This function can be used to identify periodic curves.

ts_period_detect

Syntax:

```
select ts_period_detect(x,y,minPeriod,maxPeriod)
```

The following table lists the parameters of the function.

Parameter	Description	Value

Parameter	Description	Value
x	The time sequence. The points in time along the horizontal axis are sorted in ascending order.	Each point is a UNIX timestamp. Unit: seconds.
y	The sequence of numeric data at each specified point in time.	-
minPeriod	The ratio of the minimum length of the estimated period to the total length of the time series data.	The value must be a decimal number. Value range: (0, 1].
maxPeriod	The ratio of the maximum length of the estimated period to the total length of the time series data. **Note** The value of the *maxPeriod* parameter must be greater than the value of the *minPeriod* parameter.	The parameter value must be a decimal number. Value range: (0, 1].

Example:

- Query statement

```
* | select ts_period_detect(stamp, value, 0.2, 1.0) from ( select __time__ - __time__ % 120 as stamp, avg(v) as value from log GROUP BY stamp order by stamp )
```

- Result

The following table lists the display items.

Display item	Description
period_id	An array with a length of 1. The element in the array indicates the sequence number of the period. The array [0] indicates the original time series curve.
time_series	The sequence of timestamps.
data_series	The sequence of data at each timestamp. • If the value of period_id is 0, the returned data is the original time series data. • If the value of period_id is not 0, the data returned is filtered time series data.

ts_period_classify

Syntax:

```
select ts_period_classify(stamp,value,instanceName)
```

The following table lists the parameters.

Parameter	Description	Value
stamp	The time sequence. The points in time along the horizontal axis are sorted in ascending order.	Each point in time is a UINX timestamp. Unit: seconds.
value	The sequence of numeric data at each specified point in time.	-
instanceName	The name of the time series curve.	-

Example:

- The following query statement is executed:

```
* and h : nu2h05202.nu8 | select ts_period_classify(stamp, value, name) from log
```

- Query result

line_name	prob	type
asg-2z6qjn6zf5ewg188pg5	1.0	-1.0
asg-bp1j8snc92p6v5pptgpj	0.07203669207039314	0.0
asg-wz99hse7u4ubopo5dt9o	0.0	0.0
asg-bp18oqni0gq96vy85te4	0.05590892692207093	0.0

The following table lists the display items.

Display item	Description
line_name	An array with a length of 1. The element in the array indicates the sequence number of the period. The array [0] indicates the original time series curve.
prob	The ratio of the primary period length to the length of the time series curve. Value range: [0, 1]. You can set the value to 0.15 when you perform a test.
type	The type of the curve. Valid values: -1, -2, and 0. • The value -1 indicates that the length of the time series curve is too short (less than 64 points). • The value -2 indicates the time series curve has a high fault rate (the fault rate exceeds 20%). • The value 0 indicates the time series curve is periodic.

28.1.4.9.4. Change point detection functions

This topic describes the change point detection functions in Log Service. You can use the functions to detect the change points in time series data.

The change point detection functions can detect the following two kinds of change points:

- Statistics feature changes within a specified period of time
- Anomalies in time series data

Functions

Function	Description
ts_cp_detect	Finds intervals in which data has different statistics features. The interval endpoints are change points.
ts_breakout_detect	Finds the time points at which data experiences dramatic changes.

ts_cp_detect

Syntax:

- If you cannot specify an appropriate time window size, use the following syntax. The default window size used in the function is 10.

```
select ts_cp_detect(x, y, samplePeriod)
```

- To adjust the effect specific to your business environment, you can specify the minSize parameter in the following function.

```
select ts_cp_detect(x, y, minSize)
```

The following table lists the parameters of the function.

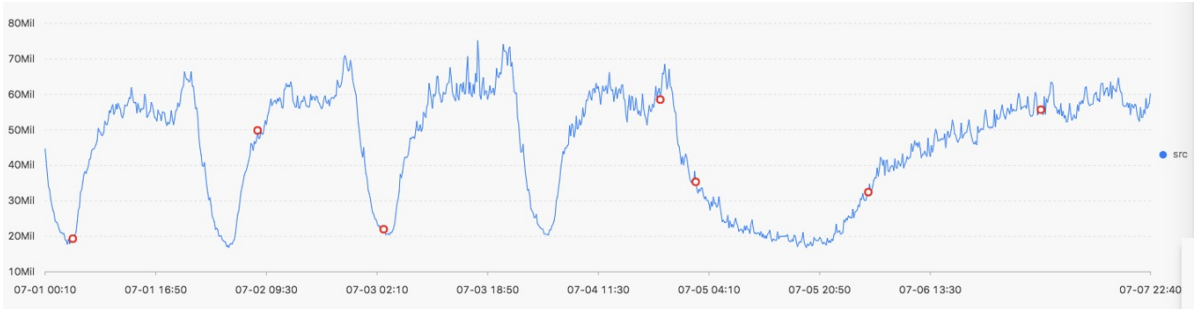
Parameter	Description	Value
x	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
y	The sequence of numeric data at each specified time point.	-
minSize	The minimum length of time series data in a continuous interval.	The minimum value is 3 and the maximum value cannot exceed ten percent of the length of the specified time series data.

Example:

- The search and analytic statement is shown as follows:

```
* | select ts_cp_detect(stamp, value, 3) from (select __time__ - __time__ % 10 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)
```

- The following figure shows the response.



The following table lists the display items.

Display item		Description
Horizontal axis	unixtime	The Unix timestamp of time series data, measured in seconds, for example, 1537071480.
Vertical axis	src	The unfiltered data, such as 1956092.7647745228.
	prob	The probability that a time point is a change point. Valid values: 0 to 1.

ts_breakout_detect

Syntax:

```
select ts_breakout_detect(x, y, winSize)
```

The following table lists the parameters of the function.

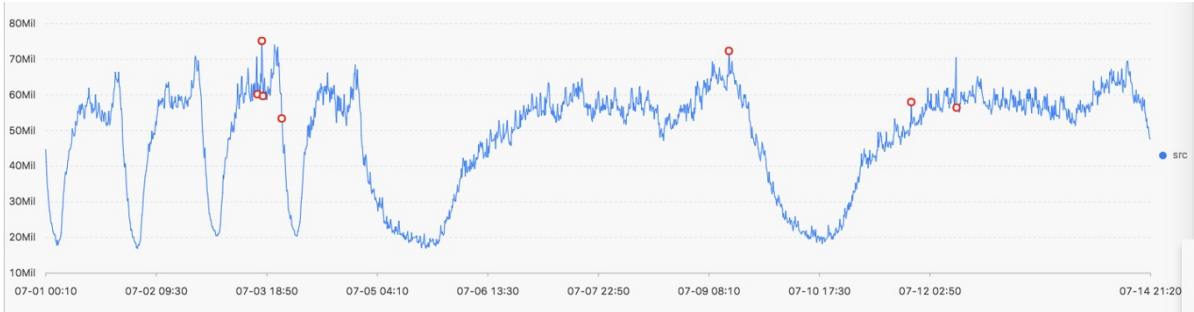
Parameter	Description	Value
x	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
y	The sequence of numeric data at each specified time point.	-
winSize	The minimum length of time series data in a continuous interval.	The minimum value is 3 and the maximum value cannot exceed ten percent of the length of the specified time series data.

Example:

- The search and analytic statement is shown as follows:

```
* | select ts_breakout_detect(stamp, value, 3) from (select __time__ - __time__ % 10 as stamp, avg
(v) as value from log GROUP BY stamp order by stamp)
```

- The following figure shows the response.



The following table lists the display items.

Display item		Description
Horizontal axis	unixtime	The Unix timestamp of time series data, measured in seconds, for example, 1537071480.
	src	The unfiltered data, such as 1956092.7647745228.

Vertical axis Display item		Description
	prob	The probability that a time point is a change point. Valid values: 0 to 1.

28.1.4.9.5. Maximum value detection function

This topic describes the available maximum value detection function in Log Service. You can use the functions to find the local maximum value of time series data in a specified window.

ts_find_peaks

Syntax:

```
select ts_find_peaks(x, y, winSize)
```

The following table lists the parameters of the function.

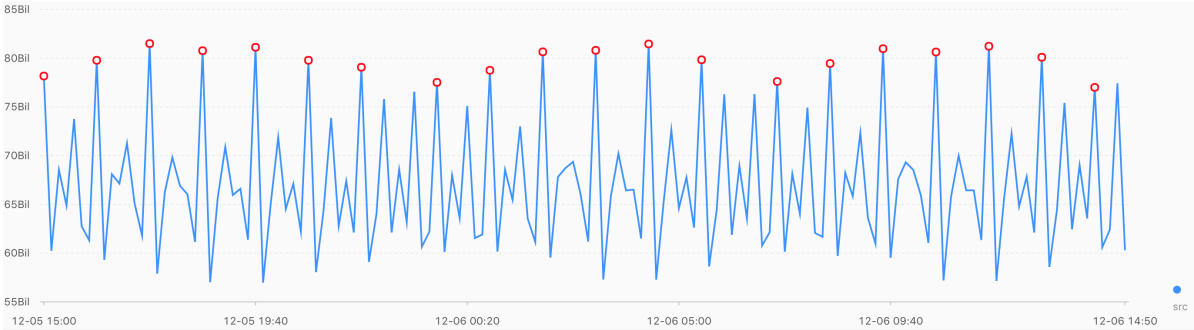
Parameter	Description	Value
x	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
y	The sequence of numeric data at each specified time point.	-
winSize	The minimum length of the detection window.	The value of the parameter is of the LONG type, ranging from 1 to the length of time series data. We recommend that you set this parameter to ten percent of the actual data length.

Example:

- The search and analytic statement is shown as follows:

```
* and h : nu2h05202.nu8 and m: NET |  select ts_find_peaks(stamp, value, 30) from (select __time__
- __time__ % 10 as stamp, avg(v) as value from log GROUP  BY  stamp order by stamp)
```

- The following figure shows the response.



The following table lists the display items.

Display item		Description
Horizontal axis	unixtime	The Unix timestamp of time series data, measured in seconds, for example, 1537071480.

Display item		Description
Vertical axis	src	The unfiltered data, such as 1956092.7647745228.
	peak_flag	Indicates whether the numeric value at the time point is the maximum value. Valid values: 1.0 and 0.0. • 1.0: The numeric value at the time point is the maximum value. • 0.0: The numeric value at the time point is not the maximum value.

28.1.4.9.6. Prediction and anomaly detection functions

Prediction and anomaly detection functions predict the trend of time series curves and identify the Ksigma and quantiles of the errors between a predicted curve and an actual curve. You can use the functions to detect anomalies.

Functions

Function	Description
ts_predicate_simple	Uses default parameters to model time series data and performs prediction and anomaly detection on time series data.
ts_predicate_ar	Uses an autoregressive (AR) model to model time series data and performs prediction and anomaly detection on time series data.
ts_predicate_arma	Uses an autoregressive moving average (ARMA) model to model time series data and performs prediction and anomaly detection on time series data.
ts_predicate_arima	Uses an autoregressive integrated moving average (ARIMA) model to model time series data and performs prediction and anomaly detection on time series data.
ts_regression_predict	Accurately predicts the trend for a periodic time series curve. Scenario: This function can be used to predict metering data, network traffic, financial data, and different business data that follows certain rules.
ts_anomaly_filter	Filters the anomalies detected from multiple time series curves based on the custom anomaly mode. The anomalies are detected during the anomaly detection. This function helps you find abnormal curves in a timely manner.

ts_predicate_simple

Syntax:

```
select ts_predicate_simple(x, y, nPred, isSmooth)
```

The following table lists the parameters of the function.

Parameter	Description	Value
x	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.

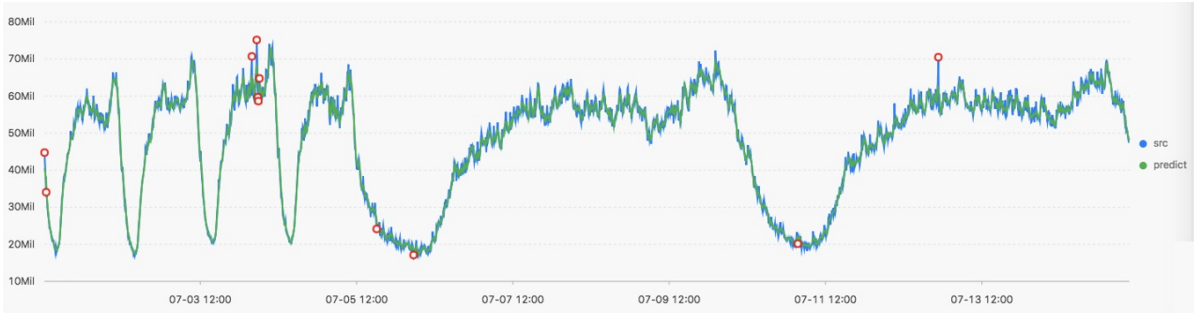
Parameter	Description	Value
y	The sequence of numeric data at each specified time point.	N/A
nPred	The number of points for prediction.	The value is of the LONG data type and must be equal to or greater than 1.
isSmooth	Specifies whether to filter the raw data.	The value is of the Boolean type. The default value is True, which indicates to filter raw data.

Example:

- A search and analytic statement is shown as follows:

```
* | select ts_predicate_simple(stamp, value, 6) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)
```

- The following figure shows the response.



The following table lists the display items.

Display item		Description
Horizontal axis	unixtime	The Unix timestamp of the data. Unit: seconds.
Vertical axis	src	The raw data.
	predict	The predicted data.
	upper	The upper limit of the prediction. The confidence level is 0.85. This value cannot be modified.
	lower	The lower limit of the prediction. The confidence level is 0.85. This value cannot be modified.
	anomaly_prob	The probability that the point is an anomaly. Valid values: [0, 1].

ts_predicate_ar

Syntax:

```
select ts_predicate_ar(x, y, p, nPred, isSmooth)
```

The following table lists the parameters of the function.

Parameter	Description	Value
x	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
y	The sequence of numeric data at each specified time point.	N/A
p	The order of the AR model.	The value is of the LONG data type. Valid values: [2, 8].
nPred	The number of points for prediction.	The value is of the LONG data type. Valid values: [1, 5 × *p*].
isSmooth	Specifies whether to filter the raw data.	The value is of the Boolean type. The default value is true, which indicates to filter raw data.

An example search and analytic statement is shown as follows:

```
* | select ts_predicate_ar(stamp, value, 3, 4) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)
```

Note The response is similar to that of the ts_predicate_simple function. For more information, see the response of the ts_predicate_simple function.

ts_predicate_arma

Syntax:

```
select ts_predicate_arma(x, y, p, q, nPred, isSmooth)
```

The following table lists the parameters of the function.

Parameter	Description	Value
x	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
y	The sequence of numeric data at each specified time point.	N/A
p	The order of the AR model.	The value is of the LONG data type. Valid values: [2, 100].
q	The order of the ARMA model.	The value is of the LONG data type. Valid values: [2, 8].
nPred	The number of points for prediction.	The value is of the LONG data type. Valid values: [*p*, 5p].
isSmooth	Specifies whether to filter the raw data.	The value is of the Boolean type. The default value is true, which indicates to filter raw data.

An example search and analytic statement is shown as follows:

```
* | select ts_predicate_arma(stamp, value, 3, 2, 4) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)
```

Note The response is similar to that of the ts_predicate_simple function. For more information, see the response of the ts_predicate_simple function.

ts_predicate_arima

Syntax:

```
select ts_predicate_arima(x,y, p, d, q, nPred, isSmooth)
```

The following table lists the parameters of the function.

Parameter	Description	Value
x	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
y	The sequence of numeric data at each specified time point.	N/A
p	The order of the AR model.	The value is of the LONG data type. Valid values: [2, 8].
d	The order of the ARIMA model.	The value is of the LONG data type. Valid values: [1, 3].
q	The order of the ARMA model.	The value is of the LONG data type. Valid values: [2, 8].
nPred	The number of points for prediction.	The value is of the LONG type. Valid values: [*p*, 5p].
isSmooth	Specifies whether to filter the raw data.	The value is of the Boolean type. The default value is True, which indicates to filter raw data.

An example search and analytic statement is shown as follows:

```
* | select ts_predicate_arima(stamp, value, 3, 1, 2, 4) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)
```

Note The response is similar to that of the ts_predicate_simple function. For more information, see the response of the ts_predicate_simple function.

ts_regression_predict

Syntax:

```
select ts_regression_predict(x, y, nPred, algotype,processType)
```

The following table lists the parameters of the function.

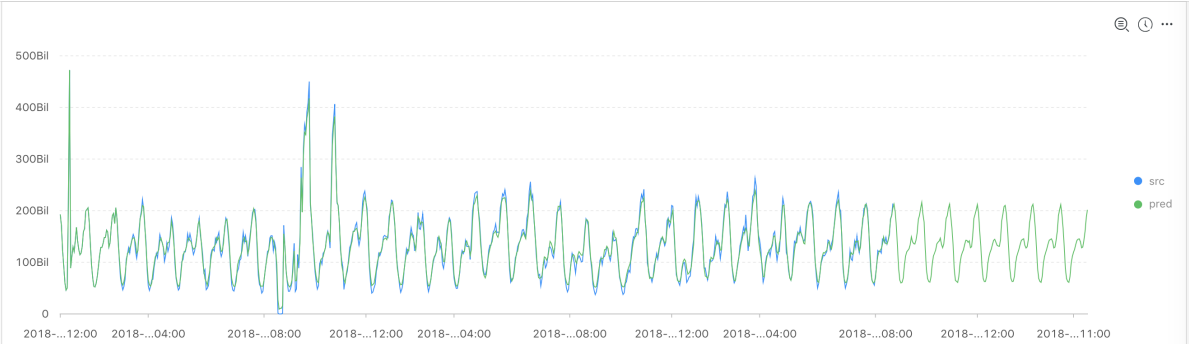
Parameter	Description	Value
x	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
y	The sequence of numeric data at each specified time point.	N/A
nPred	The number of points for prediction.	The value is of the LONG data type. Valid values: [1, 500].
algotype	The algorithm type for prediction.	Valid values: • origin: uses the Gradient Boosted Regression Tree (GBRT) algorithm for prediction. • forest: uses the GBRT algorithm for prediction based on the trend component decomposed by Seasonal and Trend decomposition using Loess (STL), and then uses the additive model to sum up the decomposed components and obtains the predicted data. • linear: uses the Linear Regression algorithm for prediction based on the trend components decomposed by STL, and then uses the additive model to sum up the decomposed components and obtains the predicted data.
processType	Specifies whether to preprocess the data.	Valid values: • 0: no additional data preprocessing is performed. • 1: removes abnormal data before prediction.

Example:

- A search and analytic statement is shown as follows:

```
* and h : nu2h05202.nu8 and m: NET |  select ts_regression_predict(stamp, value, 200, 'origin') fr
om (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GROUP  BY  stamp order by s
tamp)
```

- The following figure shows the response.



The following table lists the display items.

Display item		Description
Horizontal axis	unixtime	The Unix timestamp of the data. Unit: seconds.

Display item		Description
Vertical axis	src	The raw data.
	predict	The predicted data.

ts_anomaly_filter

Syntax:

```
select ts_anomaly_filter(lineName, ts, ds, preds, probs, nWatch, anomalyType)
```

The following table lists the parameters of the function.

Parameter	Description	Value
lineName	The name of each curve. The value is of the VARCHAR data type.	N/A
ts	The time sequence of the curve, which indicates the time of the current curve. The parameter value is an array of time points of the DOUBLE data type sorted in the ascending order.	N/A
ds	The actual value sequence of the curve. The parameter value is an array of data points with the same length as the ts parameter value.	N/A
preds	The predicted value sequence of the curve. The parameter value is an array of data points with the same length as the ts parameter value.	N/A
probs	The sequence of anomaly detection results of the curve. The parameter value is an array of data points with the same length as the ts parameter value.	N/A
nWatch	The number of the recently observed actual values on the curve. The value is of the LONG data type. The value must be smaller than the number of time points on the curve.	N/A
anomalyType	The type of anomaly to be filtered. The value is of the LONG data type.	Valid values: • 0: all anomalies. • 1: positive anomalies. • -1: negative anomalies.

Example:

- A search and analytic statement is shown as follows:

```
* | select res.name, res.ts, res.ds, res.preds, res.probs
     from (
         select ts_anomaly_filter(name, ts, ds, preds, probs, cast(5 as bigint), cast(1 as bigint)
) as res
       from (
         select name, res[1] as ts, res[2] as ds, res[3] as preds, res[4] as uppers, res[5] as low
ers, res[6] as probs
     from (
         select name, array_transpose(ts_predicate_ar(stamp, value, 10)) as res
         from (
           select name, stamp, value from log where name like '%asg-%') group by name)) );
```

- The following figure shows the response.

```
| name                     | ts                                                                | ds          |
preds     | probs       |
| ------------------------ | ----------------------------------------------------------------- | ----------- |
--------- | ----------- |
| asg-bp1hylzdi2wx7civ0ivk | [1.5513696E9, 1.5513732E9, 1.5513768E9, 1.5513804E9] | [1,2,3,NaN] |
[1,2,3,4] | [0,0,1,NaN] |
```

28.1.4.9.7. Time series decomposition function

The time series decomposition function decomposes time series curves into curves that reveal the trend and periodicity of curves.

ts_decompose

Syntax:

```
select ts_decompose(x, y)
```

The following table lists the parameters of the function.

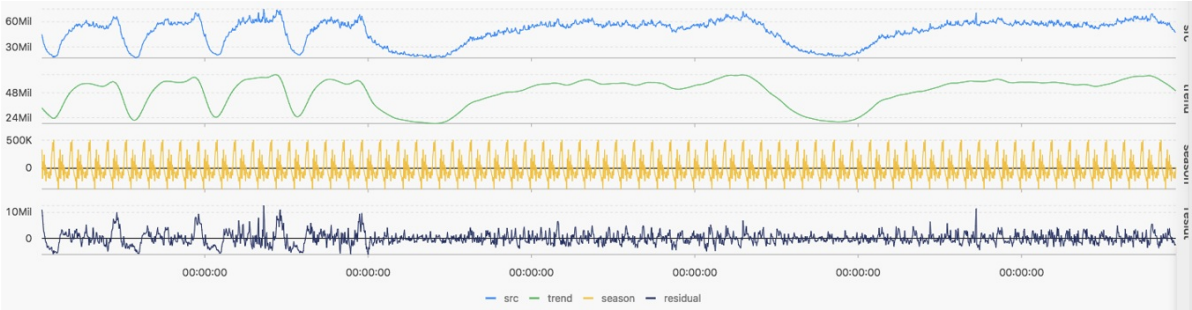
Parameter	Description	Value
x	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
y	The sequence of numeric data at each specified time point.	N/A

Example:

- A search and analytic statement is shown as follows:

```
* | select ts_decompose(stamp, value) from (select __time__ - __time__ % 60 as stamp, avg(v) as va
lue from log GROUP BY stamp order by stamp)
```

- The following figure shows the response.



The following table lists the display items.

Display item		Description
Horizontal axis	unixtime	The Unix timestamp of the data. Unit: seconds.
Vertical axis	src	The raw time series data.
	trend	The decomposed data that indicates the trend of the time series data.
	season	The decomposed data that indicates the periodicity of the time series data.
	residual	The residual data decomposed from the time series data.

28.1.4.9.8. Time series clustering functions

You can use time series clustering functions to cluster multiple time series and obtain different curve shapes. Then, you can find the cluster center and identify curves with shapes that are different from other curve shapes in the cluster in a timely manner.

Functions

Function	Description
ts_density_cluster	Uses a density-based clustering method to cluster multiple time series.
ts_hierarchical_cluster	Uses a hierarchical clustering method to cluster multiple time series.
ts_similar_instance	Queries time series curves that are similar to a specified time series curve.

ts_density_cluster

Syntax:

```
select ts_density_cluster(x, y, z)
```

The following table lists the parameters of the function.

Parameter	Description	Value

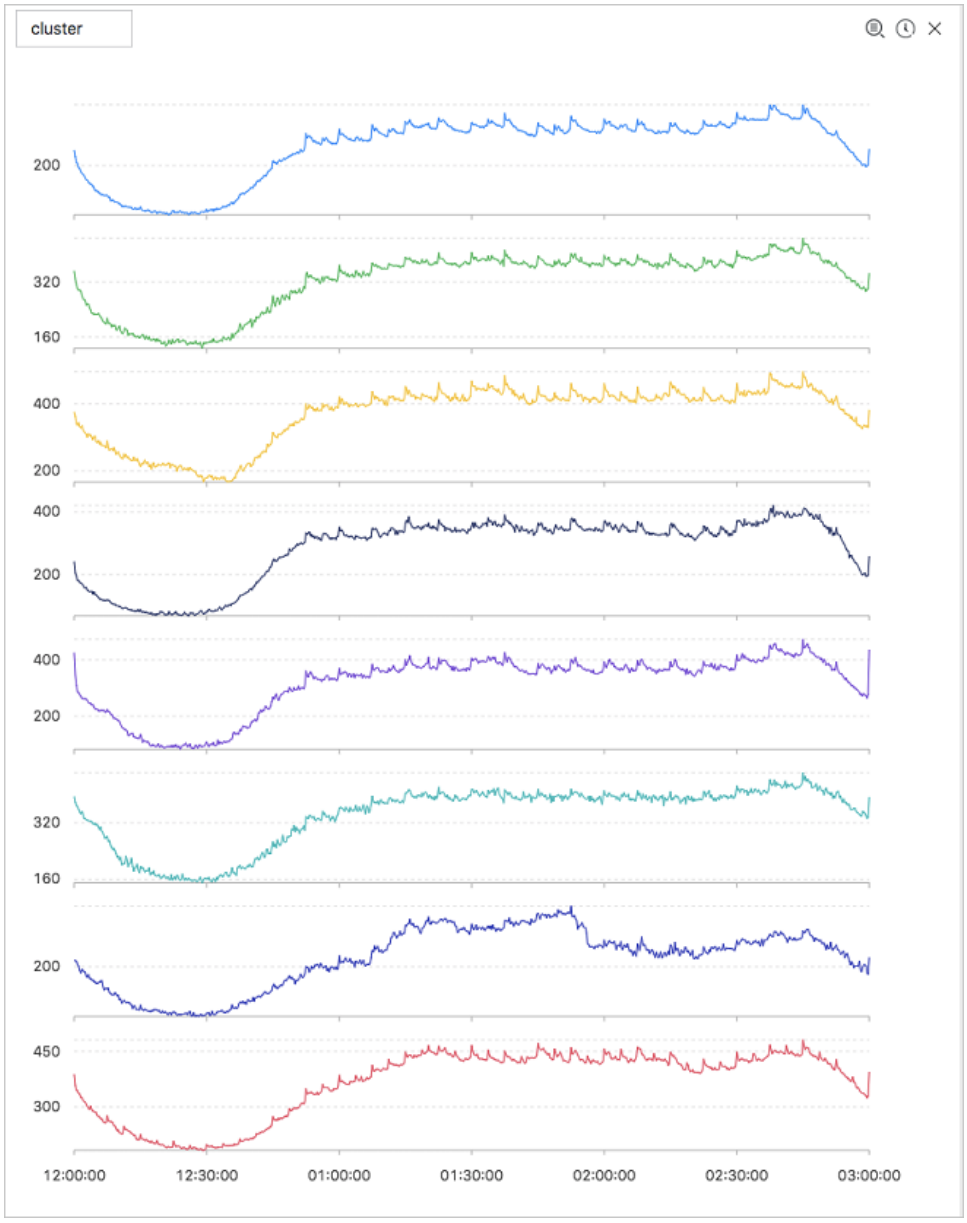
Parameter	Description	Value
x	The time sequence. The points in time along the horizontal axis are sorted in ascending order.	Each point in time is a UNIX timestamp. Unit: seconds.
y	The sequence of numeric data that corresponds to a specified point in time.	-
z	The name of the curve corresponding to the data at a specified point in time.	The value is of the string type, for example, machine01.cpu_usr.

Example

- The query statement:

```
* and (h: "machine_01" OR h: "machine_02" OR h : "machine_03") | select ts_density_cluster(stamp, metric_value,metric_name ) from ( select __time__ - __time__ % 600 as stamp, avg(v) as metric_value, h as metric_name from log GROUP BY stamp, metric_name order BY metric_name, stamp )
```

- Output result



The following table lists the display items.

Display item	Description
cluster_id	The category of the cluster. The value -1 indicates that the cluster is not categorized in any cluster center.
rate	The proportion of instances in the cluster.
time_series	The timestamp sequence of the cluster center.
data_series	The data sequence of the cluster center.
instance_names	The instances that are included in the cluster center.
sim_instance	The name of an instance in the cluster.

ts_hierarchical_cluster

Syntax:

```
select ts_hierarchical_cluster(x, y, z)
```

The following table lists the parameters of the function.

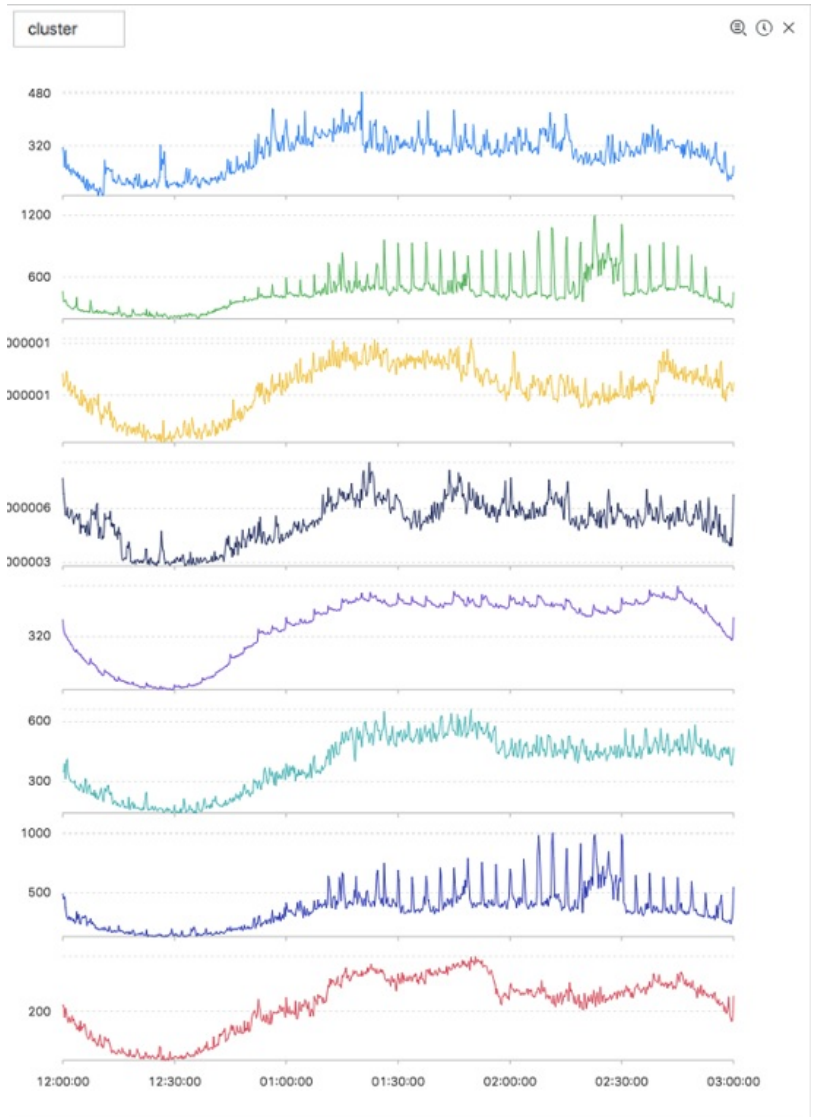
Parameter	Description	Value
x	The time sequence. The points in time along the horizontal axis are sorted in ascending order.	Each point in time is a UNIX timestamp. Unit: seconds.
y	The sequence of numeric data that corresponds to a specified point in time.	-
z	The name of the curve corresponding to the data at a specified point in time.	The value is of the string type, for example, machine01.cpu_usr.

Examples

- The query statement:

```
* and (h: "machine_01" OR h: "machine_02" OR h : "machine_03") | select ts_hierarchical_cluster(stamp, metric_value, metric_name) from ( select __time__ - __time__ % 600 as stamp, avg(v) as metric_value, h as metric_name from log GROUP BY stamp, metric_name order BY metric_name, stamp )
```

- Output result



The following table lists the display items.

Display item	Description
cluster_id	The category of the cluster. The value -1 indicates that the cluster is not categorized in any cluster center.
rate	The proportion of instances in the cluster.
time_series	The timestamp sequence of the cluster center.
data_series	The data sequence of the cluster center.
instance_names	The instances that are included in the cluster center.
sim_instance	The name of an instance in the cluster.

ts_similar_instance

Syntax:

```
select ts_similar_instance(x, y, z, instance_name, topK, metricType)
```

The following table lists the parameters of the function.

Parameter	Description	Value
x	The time sequence. The points in time along the horizontal axis are sorted in ascending order.	Each point in time is a UNIX timestamp. Unit: seconds.
y	The sequence of numeric data that corresponds to a specified point in time.	-
z	The name of the curve corresponding to the data at a specified point in time.	The value is of the string type, for example, machine01.cpu_usr.
instance_name	The name of a specified curve to be queried.	The value is of the string type, for example, machine01.cpu_usr. **Note** The curve to be queried must be an existing one.
topK	The maximum number of curves that are similar to the specified curve can be returned.	-
metricType	`{'shape', 'manhattan', 'euclidean'}`. The metric used to measure the similarity between time series curves.	-

The query statement:

```
* and m: NET and m: Tcp and (h: "nu4e01524.nu8" OR  h: "nu2i10267.nu8" OR  h : "nu4q10466.nu8") | sel
ect ts_similar_instance(stamp, metric_value, metric_name, 'nu4e01524.nu8' ) from ( select __time__ -
__time__ % 600 as stamp, sum(v) as metric_value, h as metric_name from log GROUP BY stamp, metric_nam
e order BY  metric_name, stamp )
```

The following table lists the display items.

Display item	Description
instance_name	The list of metrics that are similar to the specified metric.
time_series	The timestamp sequence of the cluster center.
data_series	The data sequence of the cluster center.

28.1.4.9.9. Frequent pattern statistics function

The frequent pattern statistics function combines representative attributes in a specified multi-attribute field sample.

pattern_stat

Syntax:

```
select pattern_stat(array[col1, col2, col3], array['col1_name', 'col2_name', 'col3_name'], array[col5
, col6], array['col5_name', 'col6_name'], support_score, sample_ratio)
```

The following table lists the parameters of the function.

Parameter	Description	Value
array[col1, col2, col3]	A column of character values.	An array of values, for example, array[clientIP, sourceIP, path, logstore].
array['col1_name', 'col2_name', 'col3_name']	The field names of the character values.	An array of field names, for example, array['clientIP', 'sourceIP', 'path', 'logstore'].
array[col5, col6]	A column of numeric values.	An array of values, for example, array[Inflow, OutFlow].
array['col5_name', 'col6_name']	The field names of the numeric values.	An array of field names, for example, array['Inflow', 'OutFlow'].
support_score	The support ratio of samples for pattern mining.	The value is of the DOUBLE data type. Value range: (0,1].
sample_ratio	The sampling ratio. The default value is 0.1, which indicates that only 10% of the total samples are used.	The value is of the DOUBLE data type. Value range: (0,1].

Example:

- Query statement

```
* | select pattern_stat(array[ Category, ClientIP, ProjectName, LogStore, Method, Source, UserAgent ], array[ 'Category', 'ClientIP', 'ProjectName', 'LogStore', 'Method', 'Source', 'UserAgent' ], array[ InFlow, OutFlow ], array[ 'InFlow', 'OutFlow' ], 0.45, 0.3) limit 1000
```

- Display item

Display item	Description
count	The number of samples in the current pattern.
support_score	The score of the current pattern. The score indicates the degree to which the current pattern is supported.
pattern	The content of the pattern. The pattern is organized in the format that is defined by the query conditions.

28.1.4.9.10. Differential pattern statistics function

The differential pattern statistics function analyzes differential patterns of specified multi-field samples based on the specified condition. It helps you identify the causes of the differences under the current condition in a timely manner.

pattern_diff

Syntax:

```
select pattern_diff(array_char_value, array_char_name, array_numeric_value, array_numeric_name, condition, supportScore,posSampleRatio,negSampleRatio )
```

The following table lists the parameters of the function.

Parameter	Description	Value
array_char_value	A column of values of the character data type.	An array of values, for example, array[clientIP, sourceIP, path, logstore].
array_char_name	The field names of the values of the character data type.	An array of field names, for example, array['clientIP', 'sourceIP', 'path', 'logstore'].
array_numeric_value	A column of values of the numeric data type.	An array of values, for example, array[Inflow, OutFlow].
array_numeric_name	The field names of the values of the numeric data type.	An array of field names, for example, array[originflow', 'OutFlow'].
condition	The condition for filtering data. The value True indicates positive samples, and the value False indicates negative samples.	For example, Latency <= 300.
supportScore	The support ratio of positive and negative samples for pattern mining.	The value is of the DOUBLE data type. Valid values: (0,1].
posSampleRatio	The sampling ratio of positive samples. The default value is 0.5, indicating that 50% of positive samples are collected.	The value is of the DOUBLE data type. Valid values: (0,1].
negSampleRatio	The sampling ratio of negative samples. The default value is 0.5, indicating that 50% of positive samples are collected.	The value is of the DOUBLE data type. Valid values: (0,1].

Example:

- A search and analytic statement is shown as follows:

```
* | select pattern_diff(array[ Category, ClientIP, ProjectName, LogStore, Method, Source, UserAgent ], array[ 'Category', 'ClientIP', 'ProjectName', 'LogStore', 'Method', 'Source', 'UserAgent' ], array[ InFlow, OutFlow ], array[ 'InFlow', 'OutFlow' ], Latency > 300, 0.2, 0.1, 1.0) limit 1000
```

- Display item

Display item	Description
possupport	The support ratio of positive samples for the mined patterns.
posconfidence	The confidence level of the mined patterns in positive samples.
negsupport	The support ratio of negative samples for the mined patterns.
diffpattern	The content of the mined patterns.

28.1.4.9.11. Root cause analysis function

Log Service provides the alert and analytics features that help you quickly analyze data and locate anomalies of specific subdimensions of a metric. You can use the root cause analysis function to analyze the subdimension attributes that result in anomalies of the monitoring metric.

rca_kpi_search

Syntax

```
select rca_kpi_search(varchar_array, name_array, real, forecast, level)
```

The following table lists the parameters of the function.

Parameter	Description	Value
varchar_array	The subdimension attributes.	The parameter value is formatted in an array, for example, array[col1, col2, col3].
name_array	The subdimension attribute names.	The parameter value is formatted in an array, for example, array['col1', 'col2', 'col3'].
real	The actual value of each subdimension attribute specified by the varchar_array parameter.	The parameter value is of the DOUBLE data type. Valid values: all real numbers.
forecast	The predicted value of each subdimension attribute specified by the varchar_array parameter.	The parameter value is of the DOUBLE data type. Valid values: all real numbers.
level	The number of subdimension attributes identified in the returned root cause set. The value 0 indicates that all root causes that are found are returned.	The parameter value is of the LONG data type. Valid values: [0, number of analyzed subdimensions]. The number of analyzed subdimensions is the length of the array specified by the varchar_array parameter.

Example:

- The search and analytic statement is shown as follows:

  Use a subquery to obtain the actual value and predicted value of each subdimension attribute, and then call the rca_kpi_search function to analyze the root causes of anomalies.

```
* not Status:200 |
select rca_kpi_search(
 array[ ProjectName, LogStore, UserAgent, Method ],
 array[ 'ProjectName', 'LogStore', 'UserAgent', 'Method' ], real, forecast, 1)
from (
select ProjectName, LogStore, UserAgent, Method,
 sum(case when time < 1552436040 then real else 0 end) * 1.0 / sum(case when time < 1552436040
then 1 else 0 end) as forecast,
 sum(case when time >=1552436040 then real else 0 end) *1.0 / sum(case when time >= 1552436040
then 1 else 0 end) as real
 from (
select __time__ - __time__ % 60 as time, ProjectName, LogStore, UserAgent, Method, COUNT(*) as rea
l
from log GROUP by time, ProjectName, LogStore, UserAgent, Method )
GROUP BY ProjectName, LogStore, UserAgent, Method limit 100000000)
```

- The following figure shows the response.

The following figure shows the structured response.

```
{
  "rcSets": [
    {
      "rcItems": [
        {
          "kpi": [("attr": "xxx", "val": "xxx")],
          "nleaf": 100,
          "change": 0.524543,
          "score": 0.1454543
        }
      ]
    }
  ]
}
```

The following table lists the display items.

Display item	Description
rcSets	The root cause sets. Each value of this parameter is an array.
rcItems	A specific root cause set.
kpi	A specific item in the root cause set. Each item is formatted in an array where each element is a JSON object. The attr parameter indicates the subdimension name, and the val parameter indicates the attribute name under the subdimension.
nleaf	The number of leaf nodes that an item (KPI) in the root cause set covers in the original data. **Note** Leaf node: the log entry that contains the finest-grained attribute information.
change	The ratio of anomalies of leaf nodes in a KPI to the total anomalies in the root cause set that occurred at the same time point.
score	The abnormality score of the current KPI. Valid values: [0, 1].

The response is formatted in a JSON object as follows:

```
{
  "rcSets": [
  {
    "rcItems": [
    {
      "kpi": [
      {
        "attr": "country",
        "val": "*"
      },
      {
        "attr": "province",
        "val": "*"
      },
      {
        "attr": "provider",
        "val": "*"
      },
      {
        "attr": "domain",
        "val": "download.huya.com"
      },
      {
        "attr": "method",
        "val": "*"
      }
      ],
      "nleaf": 119,
      "change": 0.3180687806279939,
      "score": 0.14436007709620113
    }
    ]
  }
  ]
}
```


28.1.4.9.12. Correlation analysis functions

You can use a correlation analysis function to find the metrics that are correlated with a specified metric or time series data among multiple observed metrics in the system.

Functions

Function	Description
ts_association_analysis	Quickly finds the metrics that are correlated with a specified metric among multiple observed metrics in the system.
ts_similar	Quickly finds the metrics that are correlated with specified time series data among multiple observed metrics in the system.

ts_association_analysis

Syntax

```
select ts_association_analysis(stamp, params, names, indexName, threshold)
```

The following table lists the parameters of the function.

Parameter	Description	Value
stamp	The Unix timestamp of the LONG data type.	-
params	The metrics to be analyzed, formatted in an array where each element is of the DOUBLE data type.	The parameter value is formatted in an array where each element is of the DOUBLE data type. For example, Latency, QPS, and NetFlow.
names	The names of the metrics to be analyzed.	The parameter value is formatted in an array where each element is of the VARCHAR data type. For example, Latency, QPS, and NetFlow.
indexName	The name of the target metric.	The parameter value is of the VARCHAR data type, for example, Latency.
threshold	The threshold of correlation between the metrics to be analyzed and the target metric.	The parameter value is of the DOUBLE data type. Valid values: [0, 1].

Response

- name: the name of the metric that meets the specified correlation condition with the target metric.
- score: the value of correlation between the returned metric and the target metric. Valid values: [0, 1].

Sample statement

```
* | select ts_association_analysis(
              time,
              array[inflow, outflow, latency, status],
              array['inflow', 'outflow', 'latency', 'status'],
              'latency',
              0.1) from log;
```

Sample response

```
| results               |
| --------------------- |
| ['latency', '1.0']    |
| ['outflow', '0.6265'] |
| ['status', '0.2270']  |
```

ts_similar

Syntax 1

```
select ts_similar(stamp, value, ts, ds)
select ts_similar(stamp, value, ts, ds, metricType)
```

The following table lists the parameters of the function.

Parameter	Description	Value
stamp	The Unix timestamp of the LONG data type.	-
value	The value of the metric to be analyzed. The parameter value is of the DOUBLE data type.	-
ts	The time sequence of the specified time series curve. The parameter value is formatted in an array where each element is of the DOUBLE data type.	-
ds	The sequence of numeric data of the specified time series curve.	-
metricType	The type of correlation between the measured curves. The parameter value is of the VARCHAR data type.	Valid values: SHAPE, RMSE, PEARSON, SPEARMAN, R2, and KENDALL

Syntax 2

```
select ts_similar(stamp, value, startStamp, endStamp, step, ds)
select ts_similar(stamp, value, startStamp, endStamp, step, ds, metricType )
```

The following table lists the parameters of the function.

Parameter	Description	Value
stamp	The Unix timestamp of the LONG data type.	-
value	The value of the metric to be analyzed. This parameter is of the DOUBLE data type.	-
startStamp	The start timestamp of the specified time series curve. The parameter value is of the LONG data type.	-
endStamp	The end timestamp of the specified time series curve. The parameter value is of the LONG data type.	-
step	The time interval between two adjacent data points in a time series. The parameter value is of the LONG data type.	-
ds	The sequence of numeric data of the specified time series curve. The parameter is formatted in an array where each element is of the DOUBLE data type.	-

Parameter	Description	Value
metricType	The type of correlation between the measured curves. The parameter value is of the VARCHAR data type.	Valid values: SHAPE, RMSE, PEARSON, SPEARMAN, R2, and KENDALL

- Response

  score: the correlation between the analyzed metric and the specified time series curve. Valid values: [-1, 1].

- Sample statement

```
* | select vhost, metric, ts_similar(time, value, 1560911040, 1560911065, 5, array[5.1,4.0,3.3,5.6
,4.0,7.2], 'PEARSON') from log  group by vhost, metric;
```

- Sample response

```
| vhost  | metric          | score               |
| ------ | --------------- | ------------------- |
| vhost1 | redolog         | -0.3519082537204182  |
| vhost1 | kv_qps          | -0.15922168009772697 |
| vhost1 | file_meta_write | NaN                 |
```

28.1.4.9.13. Kernel density estimation function

Kernel density estimation (KDE) is a non-parametric way to estimate the probability density function of a random variable.

The Kernel density estimation function uses the smooth peak function to fit the observed data points. In this way, the function simulates the real probability distribution curve.

- Syntax

```
select kernel_density_estimation(bigint stamp, double value, varchar kernelType)
```

- Parameters

Parameter	Description
stamp	The Unix timestamp of observed data. Unit: second.
value	The observed value.
kernelType	○ box: rectangle window. ○ epanechniov: Epanechnikov curve. ○ gausener: Gaussian curve.

- Response

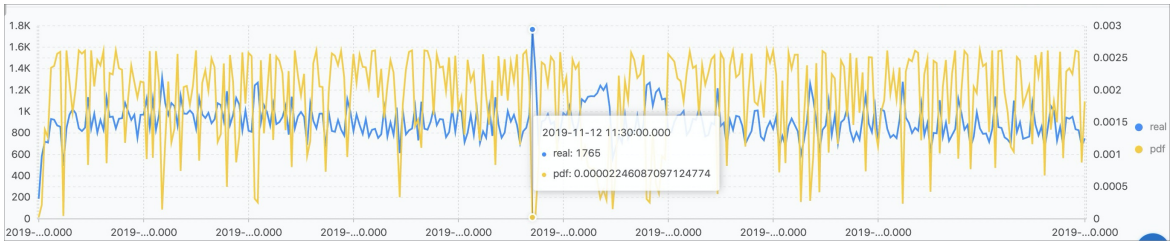
Display item	Description
unixtime	The Unix timestamp of observed data.
real	The observed value.
pdf	The probability of each observed data point.

- Example

- Sample statement

```
* |
select
    date_trunc('second', cast(t1[1] as bigint)) as time, t1[2] as real, t1[3] as pdf from (
        select kernel_density_estimation(time, num, 'gaussian') as res from (
            select __time__ - __time__ % 10 as time, COUNT(*) * 1.0 as num from log group by tim
e order by time)
        ), unnest(res) as t(t1)  limit 1000
```

- Response



28.1.4.10. Advanced analysis

28.1.4.10.1. Optimize queries

This topic describes how to optimize queries to improve query efficiency.

You can use the following methods to optimize queries:

- Increase the number of shards.
- Reduce the query time range and data volume.
- Repeat queries multiple times.
- Optimize the SQL statement for queries.

Increase the number of shards

More shards represent more computing resources and faster computing speed. You can increase the number of shards to ensure that the average number of log entries to be scanned in each shard does not exceed 50 million. You can increase the number of shards by splitting shards. For more information, see Split a shard.

> **Note** Splitting shards incurs more fees and only accelerates queries of new data. Existing data is still stored in old shards.

Reduce the query time range and data volume

- The larger the time range, the slower the query. If you query data within a year or a month, data is computed by day. To facilitate computing, you can reduce the query time range.
- The larger the data volume, the slower the query. Reduce the amount of data to be queried as much as possible.

Repeat queries multiple times

If you find that the result of a query is inaccurate, you can repeat the query multiple times. The underlying acceleration mechanism ensures that each query uses the previous query result to analyze data. In this way, multiple queries make the query result more accurate.

Optimize the SQL statement for queries

A time-consuming query statement has the following characteristics:

- Performs the GROUP BY operation on string-type columns.
- Performs the GROUP BY operation on more than five fields.
- Includes operations that generates strings.

You can use the following methods to optimize a query statement:

- Avoid operations that generate strings if possible.
  - If you use the date_format function to generate a formatted timestamp, the query is inefficient.

    ```
    * | select date_format(from_unixtime(__time__) , '%H_%i') as t, count(1) group by t
    ```

  - If you use the substr() function, strings are generated. We recommend that you use the date_trunc or time_series function in a query statement.
- Avoid performing the GROUP BY operation on string-formatted columns if possible.

  Performing the GROUP BY operation on strings may result in a large number of hash calculations, which account for more than 50% of total calculations. Examples:

  ```
  * | select count(1) as pv , date_trunc('hour',__time__) as time group by time
  * | select count(1) as pv , from_unixtime(__time__-__time__%3600) as time group by __time__-__time__%3600
  ```

  Both query 1 and query 2 count the number of log entries per hour. However, query 1 converts the time into a string, for example, 2017-12-12 00:00:00, and then performs the GROUP BY operation on this string. Query 2 calculates the on-the-hour time value, performs the GROUP BY operation on the result, and then converts the value into a string. Query 1 is less efficient than query 2 because query 1 needs to hash strings.
- List fields alphabetically based on the initial letter when performing the GROUP BY operation on multiple columns.

  For example, you need to query 100 million users who are from 13 provinces.

  ```
  Fast: * | select province,uid,count(1)groupby province,uid
  Slow: * | select province,uid,count(1)groupby uid,province
  ```

- Use estimating functions.

  Estimating functions provide stronger performance than accurate calculation. In estimation, accuracy is compromised to an acceptable extent for fast calculation.

  ```
  Fast: * |select approx_distinct(ip)
  Slow: * | select count(distinct(ip))
  ```

- Specify only required columns in the SQL statement if possible.

  You can specify all columns in the search statement. In the SQL statement, specify only required columns if possible. This will speed up calculation.

  ```
  Fast: * |select a,b c
  Slow: * |select *
  ```

- Place columns that do not need to be grouped in an aggregate function if possible.

  For example, a user ID is associated with a username. Therefore, you can execute the Group By operation on user IDs to analyze data.

  ```
  Fast: * | select userid, arbitrary(username), count(1)groupby userid
  Slow: * | select userid, username, count(1)groupby userid,username
  ```

- Avoid using the IN operator if possible.

If possible, avoid using the IN clause in SQL statements. Instead, use the OR clause.

```
Fast: key : a or key :b or key:c | select count(1)
Slow: * | select count(1) where key in ('a','b')
```

28.1.4.10.2. Use cases

This topic provides some use cases of log data analysis.

Trigger an alert when the error rate exceeds 40% over the last 5 minutes

Calculate the percentage of 500 Internal Server Error every minute. An alert is triggered when the error rate exceeds 40% over the last 5 minutes.

```
status:500 | select __topic__, max_by(error_count,window_time)/1.0/sum(error_count) as error_ratio, sum(error_count) as total_error  from (
select __topic__, count(*) as error_count , __time__ - __time__ % 300  as window_time  from log group by __topic__, window_time
)
group by __topic__  having  max_by(error_count,window_time)/1.0/sum(error_count)   > 0.4  and sum(error_count) > 500 order by total_error desc limit 100
```

Calculate the amount of transferred data and configure alerts

Calculate the amount of transferred data every minute. An alert is triggered when transferred data plunges. Transferred data counted in the last minute does not cover a full minute. The `(max(time) - min(time))` clause is used for normalization to count the average traffic per minute.

```
* | SELECT SUM(inflow) / (max(__time__) - min(__time__))  as inflow_per_minute, date_trunc('minute',__time__)  as minute group by minute
```

Calculate the average latency of traffic data in different sizes

Distribute traffic data to multiple bins based on the data size and calculate the average latency of the data in the bins.

```
* | select avg(latency) as latency , case when originSize < 5000 then 's1' when originSize < 20000 then 's2' when originSize < 500000 then 's3' when originSize < 100000000 then 's4' else 's5' end as os group by os
```

Retrieve the percentages of different results

List the number and the percentage of each result for different departments. This query includes subqueries and window functions. The `sum(c) over()` clause indicates the sum of values in all rows.

```
* |  select  department, c*1.0/ sum(c) over ()  from(select  count(1) as c, department  from log group by department)
```

Count the number of log entries that meet the query condition

To count the number of URLs based on their characteristics, you can use the CASE WHEN clause or the COUNT_IF clause. The latter clause is simpler.

```
* | select  count_if(uri like '%login') as login_num, count_if(uri like '%register') as register_num, date_format(date_trunc('minute', __time__), '%m-%d %H:%i') as time  group by time order by time limit 100
```

28.1.4.10.3. Time field conversion examples

During search and analytics, you often need to process time fields in log data, such as converting a timestamp to another time format. This topic uses some examples to describe how to convert time fields.

A log entry may include multiple time fields, for example:

- `__time__`: the time that you specify when you use the API or SDK to write log data. This field can be used for log data shipping, search, and analytics.
- Original time field in log data: the field that records the time when the log data is generated. This field is in raw logs.

Time fields in different formats are difficult to read. To simplify the read process, you can convert the time format during search and analytics. For example, you can perform the following conversions:

1. Convert __time__ to a timestamp
2. Display __time__ in a specified format
3. Convert a timestamp to a specified format

Convert __time__ to a timestamp

You can use the from_unixtime function to convert the `__time__` field to a timestamp.

```
* | select from_unixtime(__time__)
```

Display __time__ in a specified format

To display the `__time__` field in the format of `YYYY-MM-DD HH:MM:SS`, you can use the date_format function.

```
* | select date_format(__time__, '%Y-%m-%d %H:%i:%S')
```

Convert the time in a log to a specified format

To convert the time field in a log to the specified format (`YYYY-MM-DD HH:MM:SS`) and perform the GROUP BY operation on the `YYYY-MM-DD` part, you can use the date_format function.

- Sample log entry

```
__topic__:
body_byte_sent:  307
hostname:  www.host1.com
http_user_agent:  Mozilla/5.0 (iPhone; CPU iPhone OS 10_3_3 like Mac OS X) AppleWebKit/603.3.8 (KH
TML, like Gecko) Mobile/14G60 QQ/7.1.8.452 V1_IPH_SQ_7.1.8_1_APP_A Pixel/750 Core/UIWebView NetTyp
e/WIFI QBWebViewType/1
method:  GET
referer:  www.host0.com
remote_addr:  36.63.1.23
request_length:  111
request_time:  2.705
status:  200
upstream_response_time:  0.225582883754
url:  /? k0=v9&
time:2017-05-17 09:45:00
```

- Example SQL statement

```
* | select date_format (date_parse(time,'%Y-%m-%d %H:%i:%S'), '%Y-%m-%d') as day, count(1) as uv g
roup by day order by day asc
```

28.1.4.11. Visual analysis

28.1.4.11.1. Analysis graph

28.1.4.11.1.1. Overview

All search and analytics results can be rendered by using visualized charts.

Prerequisites

- The index feature is enabled and configured. The analytics switches are turned on. For more information, see Enable the index feature and configure indexes for a Logstore.
- An analytic statement is included in a query statement. You cannot use charts to show query results if you do not include an analytic statement in your query statement.

Precautions

When multiple search and analytic statements are being executed in sequence, the **Value Column**, **X Axis**, or **Y Axis** information cannot automatically change based on the search and analytic statement. The X and Y axis information may remain the same as the last search and analytic statement. If this happens, the query results of the current search and analytic statement cannot be automatically displayed in a chart. If the following messages are returned, configure parameters on the **Properties** tab based on the current search and analytic statement:

- The currently selected dimensions are not in the queried results. Check and configure the attributes.
- X-Axis or Y-Axis is not available. Check and configure the attributes.

Chart configurations

On the **Graph** tab, various charts are provided to show query results. You can select a type of chart from the chart bar to show results.

- On the **Graph** tab, you can view the **Chart Preview** and **Data Preview** of query results of the current search and analytic statement. **Chart Preview** is the preview of the query results that are displayed in the specified type of chart. **Data Preview** displays the query results in a table.
- On the **Graph** tab on the right, you can configure the following chart properties:
  - **Data Source**: used to set placeholder variables. For example, you configure the drill-down event of Chart A to redirect to the dashboard where Chart B is located. The placeholder variable you configured for Chart B is the same as the variable that you click to trigger the drill-down event. Then the placeholder variable is replaced with the variable you click to trigger the drill-down event and the search and analytic statement of Chart B is executed. For more information, see Drill-down analysis.

    This feature is applicable to scenarios where you configure drill-down events to redirect to targeted dashboards.
  - **Properties**: used to configure the display properties of a chart, including the X axis, left and right Y axes, margins, font size and other properties. The properties vary with different type of charts.

    This feature is applicable to all search and analytics scenarios.
  - **Interactive Behavior**: used to configure drill-down events for a chart. After you configure a drill-down event for the chart, you can click the variable value in the chart to trigger the specified drill-down event. For more information, see Drill-down analysis.

    This feature is applicable to triggering drill-down events for charts.

28.1.4.11.1.2. Display query results on a table

Tables are used to sort and display data for quick reference and analysis. All query results that match specified query statements can be rendered into visualized charts. By default, the query results are displayed in a table.

Components

- Table header
- Row
- Column

Where:

- The number of columns can be specified by using a `SELECT` statement.
- The number of rows is calculated based on the number of log entries in a specified time range. The default clause is `LIMIT 100`.

Procedure

1. On the Search & Analysis page, enter a query statement in the search box, specify a time range, and then click **Search & Analyze**.
2. On the **Graph** tab, data is displayed in a table by default. You do not need to click the  icon.
3. On the **Properties** tab, configure the properties of the table.

Properties

Parameter	Description
Items per Page	The number of log entries to return on each page.
Zebra Striping	Specifies whether to display the query results in a zebra-striped table.
Transpose Rows and Columns	Specifies whether to transpose rows and columns.
Hide Reserved Fields	Specifies whether to hide reserved fields.
Disable Sorting	Specifies whether to disable the sorting feature.
Disable Search	Specifies whether to disable the search feature.
Highlight Settings	The rules for highlighting rows or columns that conform to specified rules.

28.1.4.11.1.3. Display query results on a line chart

A line chart is used to analyze the value changes of fields based on an ordered data type. In most cases, this analysis is based on a specified time range.

You can use a line chart to analyze the following change characteristics of field values over a specified period:

- Increment or decrement
- Increment or decrement rate
- Increment or decrement pattern, for example, periodicity
- Peak value and trough value

Line charts are used to analyze field value changes over a time period. You can also use a line chart to analyze the value changes of multiple fields in multiple lines over the same time period. Then, you can analyze the relationships between the different fields. For example, the values of multiple fields can display positive, negative, or inverse trends.

Components

- X-axis
- Left Y-axis
- Right Y-axis (optional)
- Data point
- Line of trend change
- Legend

Procedure

1. On the Search & Analysis page, enter a query statement in the search box, specify a time range, and then click **Search & Analyze**.
2. On the **Graph** tab, click the  icon.
3. On the **Properties** tab, configure the properties of the line chart.

> **Note** In a line chart, each line must contain more than two data points. Otherwise, the data trend cannot be analyzed. We recommend that you select five or fewer lines in a line chart.

Properties

Parameter	Description
X Axis	The sequential data. In most cases, time series is selected.
Left Y Axis	The numeric data. You can select one or more fields for the left Y-axis.
Right Y Axis	The numeric data. You can select one or more fields for the right Y-axis. The layer of the right Y-axis is higher than that of the left Y-axis.
Column Marker	The column on the left or right Y-axis. The column is selected as a histogram.
Legend	The position where the legend is located in the chart. Valid values: Top, Bottom, Left, and Right.
Format Left Y-axis	The format in which data selected for the left Y-axis and right Y-axis is displayed.
Format Right Y-axis	
Margin	The distance between an axis and the borders of the chart. Valid values: **Top Margin**, **Bottom Margin**, **Right Margin**, and **Left Margin**.

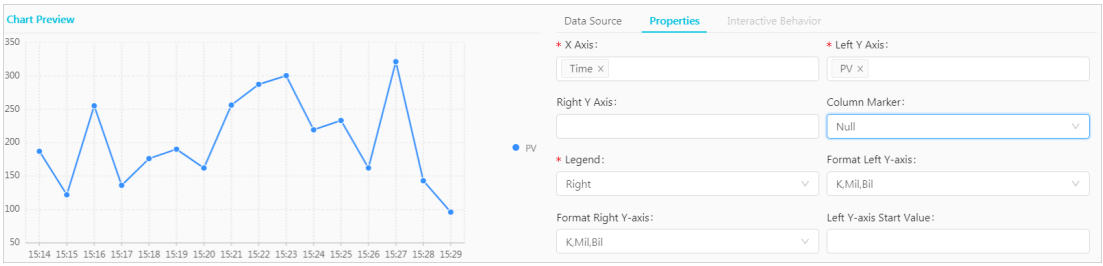
Example of a line chart

To query the page views (PVs) of the IP address `10.0.192.0` in the last 24 hours, execute the following query statement:

```
remote_addr: 10.0.192.0 | select date_format(date_trunc('hour', __time__), '%m-%d %H:%i')
as time, count(1) as PV group by time order by time limit 1000
```

Select `time` for the X-axis, `PV` for the left Y-axis, and Bottom for Legend. Adjust the margins based on your business requirements.

Line chart



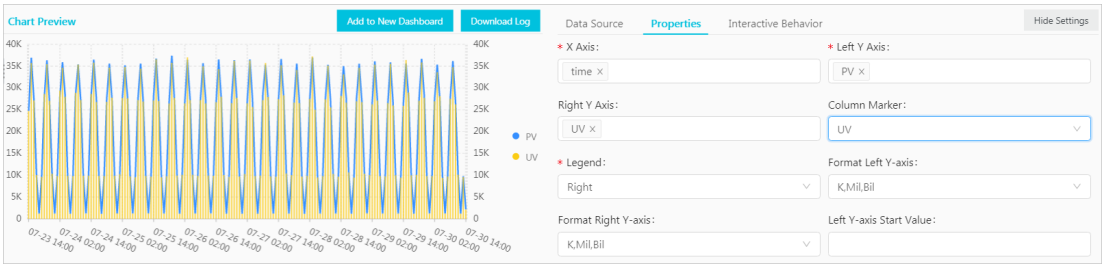
Example of a dual Y-axis line chart

To query the PVs and unique visitors (UVs) in the last 24 hours, execute the following query statement:

```
* | select date_format(date_trunc('hour', __time__), '%m-%d %H:%i') as time, count(1) as PV, approx_d
istinct(remote_addr) as UV group by time order by time limit 1000
```

Select `time` for the X-axis, `PV` for the left Y-axis, `UV` for the right Y-axis, and `PV` for Column Marker.

Dual Y-axis line chart



28.1.4.11.1.4. Display query results on a column chart

A column chart uses vertical or horizontal bars to present categorical values. Compared with a line chart, a column chart does not display ordered data, but provides a method to count the number of values in each category.

Components

- X-axis (horizontal)
- Y-axis (vertical)
- Rectangular bar
- Legend

By default, column charts in Log Service use vertical bars. Each rectangular bar has a fixed width and a variable height that indicates a value. You can use a grouped column chart to display the data if multiple columns of data are mapped to the Y-axis.

Procedure

1. On the Search & Analysis page, enter a query statement in the search box, specify a time range, and then click

Search & Analyze.

2. On the **Graph** tab, click the icon.
3. On the **Properties** tab, configure the properties of the column chart.

> **Note** Column charts can be used to display query results if the number of returned log entries is less than 20. You can use a `LIMIT` clause to control the number of rectangular bars. Analysis results may not be clearly displayed if the chart contains a large number of rectangular bars. In addition, we recommend that you select less than five fields for the Y-axis.

Properties

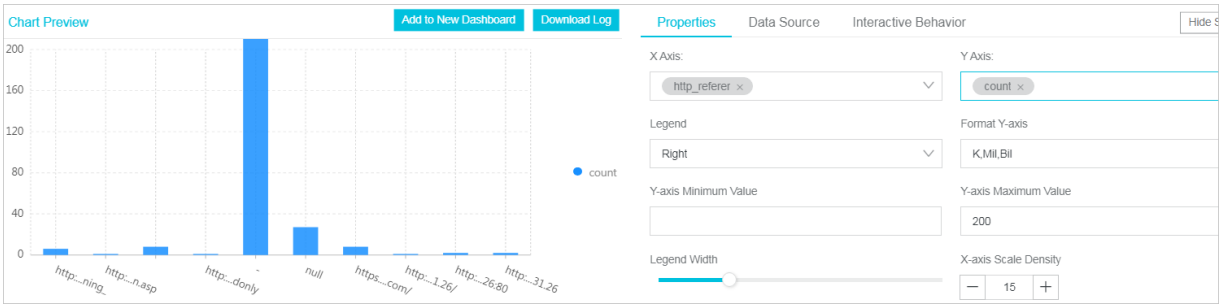
Parameter	Description
X Axis	The categorical data.
Y Axis	The numeric data. You can select one or more fields for the Y-axis.
Legend	The position where the legend is located in the chart. Valid values: Top, Bottom, Left, and Right.
Format	The format in which data is displayed on the Y-axis.
Margin	The distance between an axis and the borders of the chart. Valid values: **Top Margin**, **Bottom Margin**, **Right Margin**, and **Left Margin**.

Example of a simple column chart

To query the number of visits for each `http_referer` in the specified time range, execute the following query statement:

```
* | select  http_referer, count(1) as count group by http_referer
```

Select `http_referer` for X-axis and `count` for Y-axis.



Example of a grouped column chart

To query the number of requests and the average bytes for each `http_referer` in the specified time range, execute the following statement:

```
* | select  http_referer, count(1) as count, avg(body_bytes_sent) as avg group by http_referer
```

Select `http_referer` for X-axis. Select `count` and `avg` for Y-axis.

28.1.4.11.1.5. Display query results on a bar chart

A bar chart is a horizontal column chart that is used to analyze the top N values of fields. A bar chart is configured in a similar way to a column chart.

Components

- X-axis (vertical)
- Y-axis (horizontal)
- Rectangular bar
- Legend

Each rectangular bar has a fixed height and a variable width. The variable width indicates a value. You can use a grouped bar chart to display the data if multiple columns of data are mapped to the Y-axis.

Procedure

1.
2. On the **Graph** tab, click the icon.
3. On the **Properties** tab, configure the properties of the bar chart.

> **Note**
>
> - Bar charts can be used to display query results if 20 or fewer log entries are returned. You can use the `LIMIT` clause to control the number of rectangular bars. Analysis results may not be clearly displayed if the chart contains a large number of rectangular bars. You can use the `ORDER BY` clause to analyze the top N values of fields. We recommend that you map less than five columns of data to the Y-axis.
> - You can use a grouped bar chart to display query results. However, the values represented by each rectangular bar in a group must be positively or negatively associated with each other.

Properties

Parameters

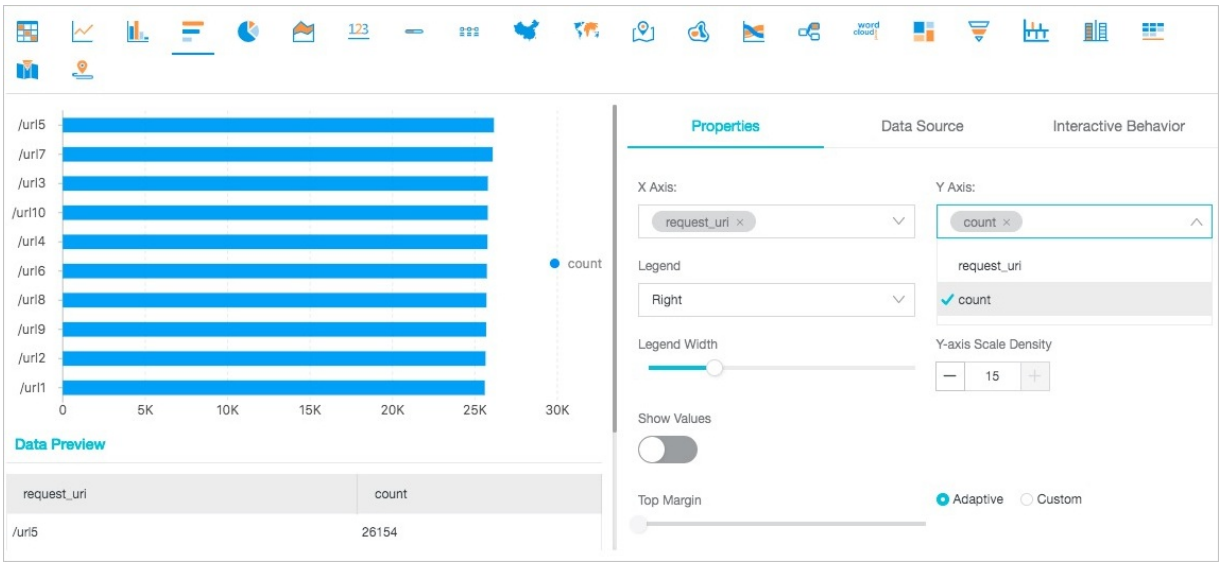
Parameter	Description
X Axis	The categorical data.
Y Axis	The numeric data. You can select one or more fields for the Y-axis.

Parameter	Description
Legend	The position where the legend is located in the chart. Valid values: Top, Bottom, Left, and Right.
Format X-axis	The format in which data is displayed on the X-axis.
Margin	The distance between an axis and the borders of the chart. Valid values: **Top Margin**, **Bottom Margin**, **Right Margin**, and **Left Margin**.

Examples

To analyze the top 10 visited request URIs (`request_uri`), execute the following query statement:

```
* | select  request_uri, count(1) as count group by request_uri order by count desc limit 10
```



28.1.4.11.1.6. Display query results on a pie chart

A pie chart is used to indicate the percentages of different data types and compare different data types based on the arc length of each slice.

Components

- Segment
- Percentage in the text format
- Legend

Types

Log Service provides three types of pie charts: a standard pie chart, donut chart, and polar area chart.

- Standard pie chart

  A standard pie chart is divided into multiple segments based on the percentages of various field values. The entire chart displays all field values. Each segment displays the percentage and the numeric value of a field. The sum of percentages from all segments is equal to 100%.

- Donut chart

A donut chart is a standard pie chart with a hollow center. A donut chart has the following benefits:

- In addition to the information that a standard pie chart displays, a donut chart displays the total number of occurrences of all field values.
- You can view the differences between the number of occurrences of the same value in two charts based on the ring length. This is more intuitive than comparing two standard pie charts.

- Polar area chart

  A polar area chart is a column chart in the polar coordinate system. Each category of field values is represented by a segment with the same radian. The radius of a segment indicates the number of occurrences of a field value. Compared with a standard pie chart, a polar area chart has the following benefits:

  - Standard pie charts are suitable to display query results if 10 or fewer log entries are returned. Polar area charts are suitable for displaying query results if the number of returned log entries ranges from 10 to 30.
  - The area is the square of a radius. Therefore, the display of the polar area chart enlarges the differences among multiple types of data. This is best suited for a comparison of similar values.
  - A circle can be used to display a periodic pattern. Therefore, you can use a polar area chart to analyze value change characteristics in specified periods, such as weeks and months.

Procedure

1.
2. On the **Graph** tab, click the  icon.
3. On the **Properties** tab, configure the properties of the pie chart.

> **Note**
>
> - Donut charts and standard pie charts can be used to display query results if less than 10 log entries are returned. You can use a `LIMIT` clause to control the number of segments. Analysis results may not be clearly displayed if the chart contains a large number of segments of different colors.
> - We recommend that you use a polar area chart or column chart if the number of returned log entries exceeds 10.

Properties

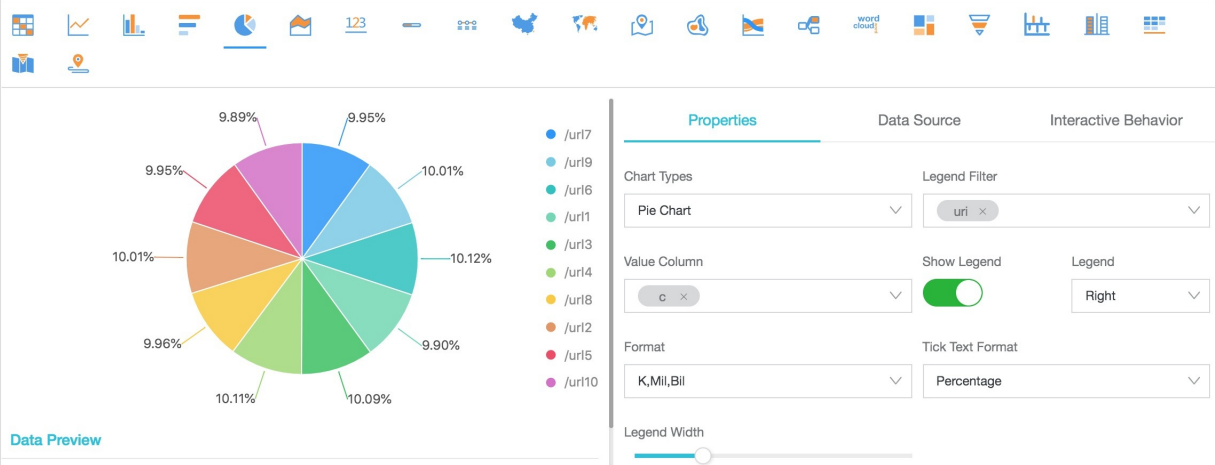
Parameter	Description
Chart Types	The type of the chart. Valid values: Pie Chart, Donut Chart, and Polar Area Chart. Default value: Pie Chart.
Legend Filter	The categorical data.
Value Column	The values that correspond to different types of data.
Show Legend	Specifies whether to show the legend.
Legend	The position where the legend is located in the chart. Valid values: Top, Bottom, Left, and Right. You can configure this parameter only after you turn on the Show Legend switch.
Format	The format in which data is displayed.
Tick Text Format	The format of the tick.

Parameter	Description
Margin	The distance between an axis and the borders of the chart. Valid values: **Top Margin**, **Bottom Margin**, **Right Margin**, and **Left Margin**.

Example of a standard pie chart

To analyze the percentages of the `request_uri` field values, execute the following query statement:

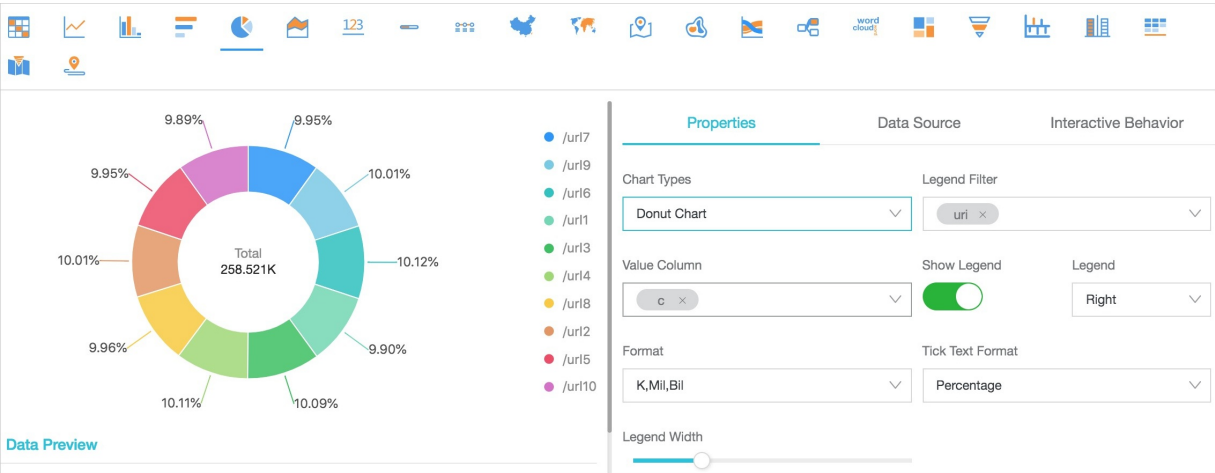
```
* | select requestURI as uri , count(1) as c group by uri limit 10
```



Example of a donut chart

To analyze the percentages of the `request_uri` field values, execute the following query statement:

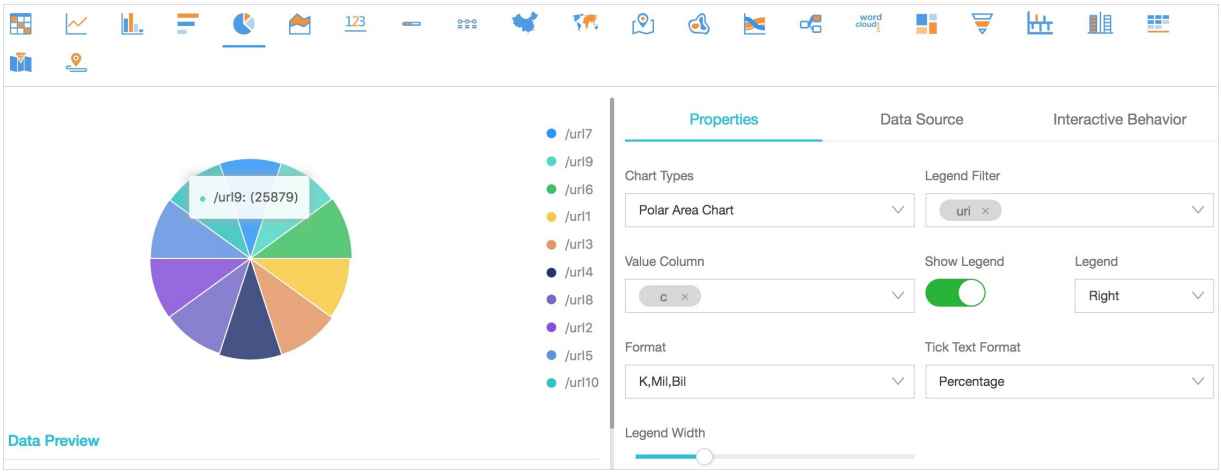
```
* | select requestURI as uri , count(1) as c group by uri limit 10
```



Example of a polar area chart

To analyze the percentages of the `request_uri` field values, execute the following query statement:

```
* | select requestURI as uri , count(1) as c group by uri limit 10
```



28.1.4.11.1.7. Display query results on an area chart

An area chart is built based on a line chart. The colored section between a line and the axis is an area. The color is used to highlight the trend. Similar to a line chart, an area chart shows the numeric value changes over a specified time period to highlight the overall data trend. Both the line chart and the area chart display the trend and relationship between numeric values instead of displaying specific values.

Components

- X-axis (horizontal)
- Y-axis (vertical)
- Area segment

Procedure

1.
2. On the **Graph** tab, click the icon.
3. On the **Properties** tab, configure the properties of the area chart.

> **Note** In an area chart, a single area segment must contain more than two data points. Otherwise, the data trend cannot be analyzed. We recommend that you select five or fewer area segments in an area chart.

Properties

Parameter	Description
X Axis	The sequential data. In most cases, time series is selected.
Y Axis	The numeric data. You can select one or more fields for the Y-axis.
Legend	The position where the legend is located in the chart. Valid values: Top, Bottom, Left, and Right.
Format	The format in which data is displayed.

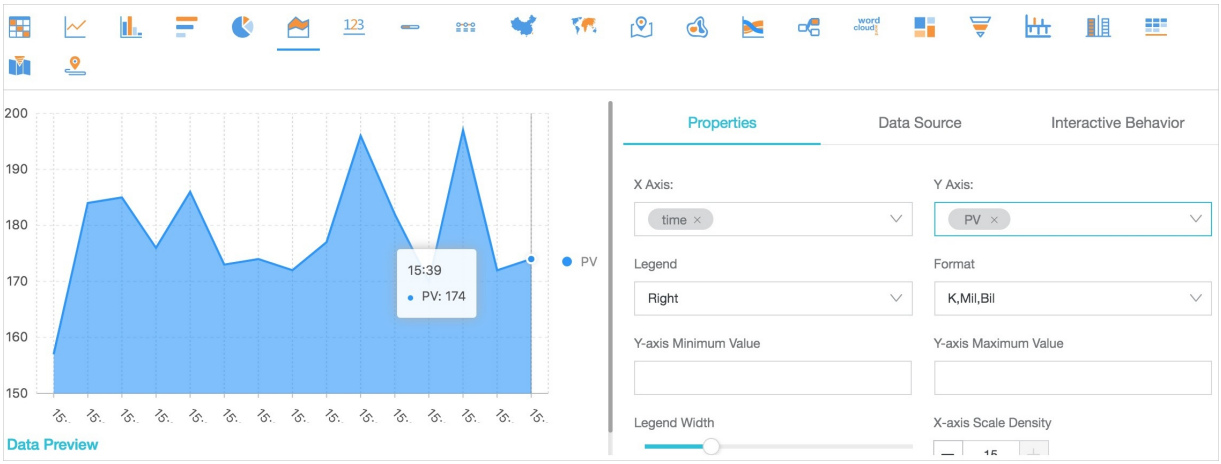
Parameter	Description
Margin	The distance between an axis and the borders of the chart. Valid values: **Top Margin**, **Bottom Margin**, **Right Margin**, and **Left Margin**.

Example of a simple area chart

To query the page views (PVs) of the IP address `10.0.192.0` in the last 24 hours, execute the following query statement:

```
remote_addr: 10.0.192.0 | select date_format(date_trunc('hour', __time__), '%m-%d %H:%i') as time, co
unt(1) as PV group by time order by time limit 1000
```

Select `time` for the X-axis and `PV` for the Y-axis.



Example of a cascade chart

```
* | select date_format(date_trunc('hour', __time__), '%m-%d %H:%i') as time, count(1) as PV, approx_d
istinct(remote_addr) as UV group by time order by time limit 1000
```

Select `time` for the X-axis. Select `PV` and `UV` for the Y-axis.

28.1.4.11.1.8. Display query results on a single value chart

A single value chart displays a single value.

Single value charts have the following types:

- Rectangle Frame: shows a general value.
- Dial: shows the difference between the current value and the specified threshold value.
- Compare Numb Chart: shows the SQL query results of interval-valued comparison and periodicity-valued comparison functions. For more information about the analytic syntax, see Interval-valued comparison and periodicity-valued comparison functions.

Rectangle Frame is selected by default. A rectangle frame is the most basic method to display data at a specified point. In most cases, it is used to show the key information at a specified point in time. To display a proportional metric, you can select Dial.

Components

- Numeric value
- Chart type

Procedure

1.
2. On the **Graph** tab, click the 123 icon.
3. On the **Properties** tab, configure the properties of the single value chart.

> **Note** Log Service normalizes data in numeric value-based charts. For example, `230000` is normalized to `230K`. You can include Mathematical calculation functions in query statements to customize numeric formats.

Properties

- The following table lists the parameters of a rectangle frame.

Parameter	Description
Chart Types	The type of the chart. Select Rectangle Frame.
Value Column	The value displayed in the chart. By default, data in the first row of the specified column is displayed.
Unit	The unit of data.
Unit Font Size	The font size of the unit. You can drag the slider to adjust the font size. Valid values: 10 to 100. Unit: pixels.
Description	The description of the value.
Description Font Size	The font size of the value description. You can drag the slider to adjust the font size. Valid values: 10 to 100. Unit: pixels.
Format	The format in which data is displayed.
Font Size	The font size of the value. You can drag the slider to adjust the font size. Valid values: 10 to 100. Unit: pixels.
Font Color	The color of the value, unit, and description in the chart. You can use the default color or select a color.

Parameter	Description
Background Color	The color of the background. You can use the default color or select another color.

- The following table describes the parameters of a dial.

Parameter	Description
Chart Types	The type of the chart. Select **Dial** to display query results on a dial.
Actual Value	The actual value in the chart. By default, data in the first row of the specified column is displayed.
Unit	The unit of the value on the dial.
Font Size	The font size of the value and unit. Valid values: 10 to 100. Unit: pixels.
Description	The description of the value.
Description Font Size	The font size of the value description. You can drag the slider to adjust the font size. Valid values: 10 to 100. Unit: pixels.
Dial Maximum	The maximum value of the scale in the dial. Default value: 100.
Maximum Value Column	The maximum value in the specified column. If you turn on the Use Query Results switch, **Dial Maximum** is replaced by Maximum Value Column. Then, you can select the maximum value from query results for this parameter.
Use Query Results	If you turn on the Use Query Results switch, Dial Maximum is replaced by Maximum Value Column. Then, you can select the maximum value from query results for this parameter.
Format	The format in which data is displayed.
Colored Regions	The number of segments that divide the dial. Each segment is displayed in a different color. Valid values: 2, 3, 4, and 5. Default value: 3.
Region Max Value	The maximum value of the scale in each colored segment of the dial. By default, the maximum value in the last segment is the maximum value on the dial. You do not need to specify this value. **Note** A dial is evenly divided into three colored segments by default. If you change the value of **Colored Regions**, Region Max Value is not automatically adjusted. You can manually set the maximum value for each colored segment based on your business requirements.
Font Color	The color of the value on the dial.
Region	The colored segments that divide the dial. A dial is evenly divided into three segments by default. The segments are displayed in blue, yellow, and red. If you set **Colored Regions** to a value greater than 3, the added segments are displayed in blue by default. You can change the color of each segment.

Parameter	Description
Show Title	Specifies whether to display the title of a dial when you add the dial to a dashboard. The **Show Title** feature can show or hide the title of the dial on the dashboard page. This switch is turned off by default. If you turn on the Show Title switch, the title of the dial is not displayed on the current page. You must create or modify a dashboard and view the title on the dashboard page.

- The following table lists the parameters of a compare numb chart.

Parameter	Description
Chart Types	The type of the single value chart. If you select Compare Numb Chart, query results are displayed on a compare numb chart.
Show Value	The value that is displayed in the center of the compare numb chart. In most cases, this value is set to the statistical result that is calculated by the related comparison function in the specified time range.
Compare Value	The value that is compared with the threshold. Set the value to the result of the comparison between the statistical results calculated by the related comparison function in the specified time range and the previously specified time range.
Font Size	The font size of the show value. Valid values: 10 to 100. Unit: pixels.
Unit	The unit of the show value.
Unit Font Size	The font size of the unit for the show value. Valid values: 10 to 100. Unit: pixels.
Compare Unit	The unit of the compare value.
Compare Font Size	The font size of the compare value and unit. Valid values: 10 to 100. Unit: pixels.
Description	The description of the show value and its growth trend.
Description Font Size	The font size of the description. Valid values: 10 to 100. Unit: pixels.
Trend Comparison Threshold	The value that is used to measure the variation trend of the compare value. For example, the compare value is -1. ◦ If you set **Trend Comparison Threshold** to 0, a down arrow that indicates a value decrease is displayed on the page. ◦ If you set **Trend Comparison Threshold** to -1, it indicates that the value remains unchanged. The system does not display the trend on the page. ◦ If you set **Trend Comparison Threshold** to -2, an up arrow that indicates a value increase is displayed on the page.
Format	The format in which data is displayed.
Font Color	The color of the show value and its description.
Growth Font Color	The font color of the compare value that is greater than the threshold.

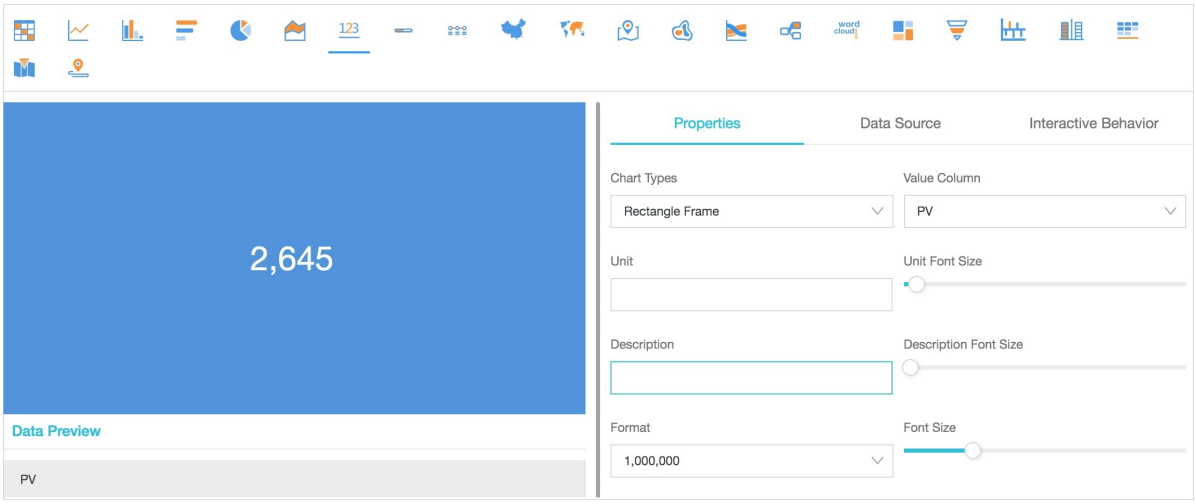
Parameter	Description
Growth Background Color	The background color displayed when the compare value is greater than the threshold.
Decrease Font Color	The font color displayed when the compare value is less than the threshold.
Decrease Background Color	The background color displayed when the compare value is less than the threshold.
Equal Background Color	The background color displayed when the compare value is equal to the threshold.

Examples

To view the number of access requests, execute the following query statements:

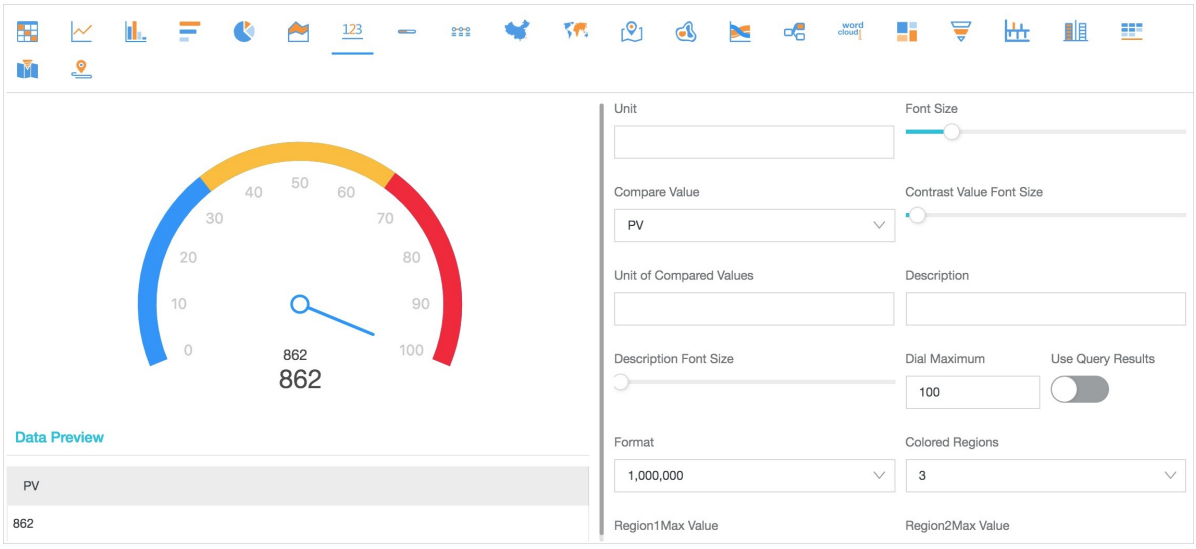
- Rectangle frame

```
* | select count(1) as pv
```



- Dial

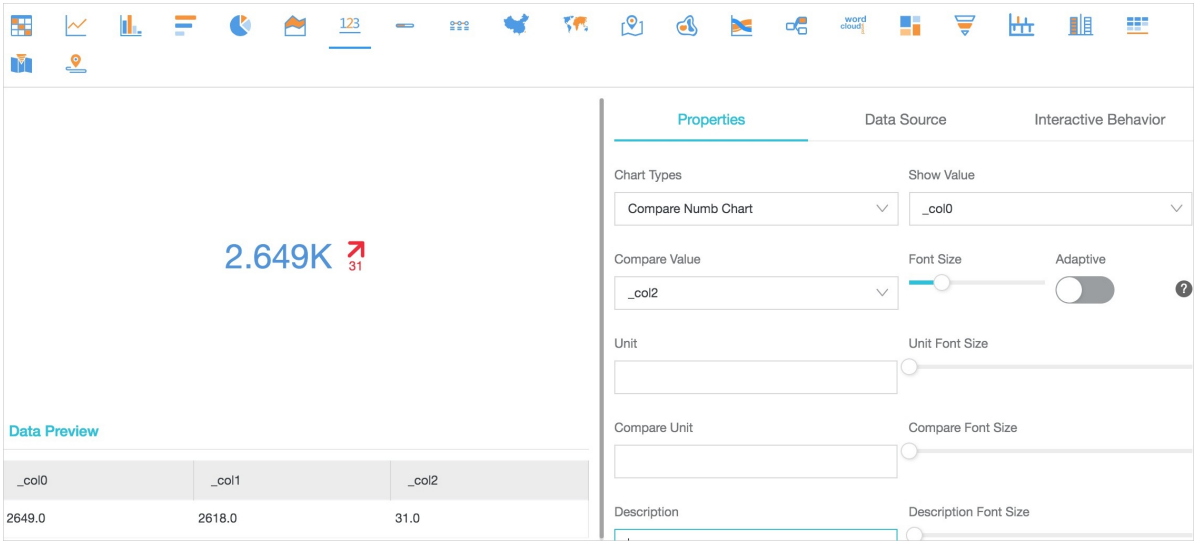
```
* | select count(1) as pv
```



- Compare numb chart

  To view and compare the access requests for today and yesterday, execute the following query statement:

```
* | select diff[1],diff[2], diff[1]-diff[2] from (select compare( pv , 86400) as diff from (select count(1) as pv from log))
```



28.1.4.11.1.9. Display query results on a progress bar

The progress bar shows the percentage of the actual value of a field to the maximum value of the field. You can configure the properties of the progress bar to adjust its style and set display rules.

Components

- Actual value
- Unit (optional)
- Total value

Procedure

1.
2. On the **Graph** tab, click to select the progress bar.
3. Configure the properties of the progress bar.

Properties

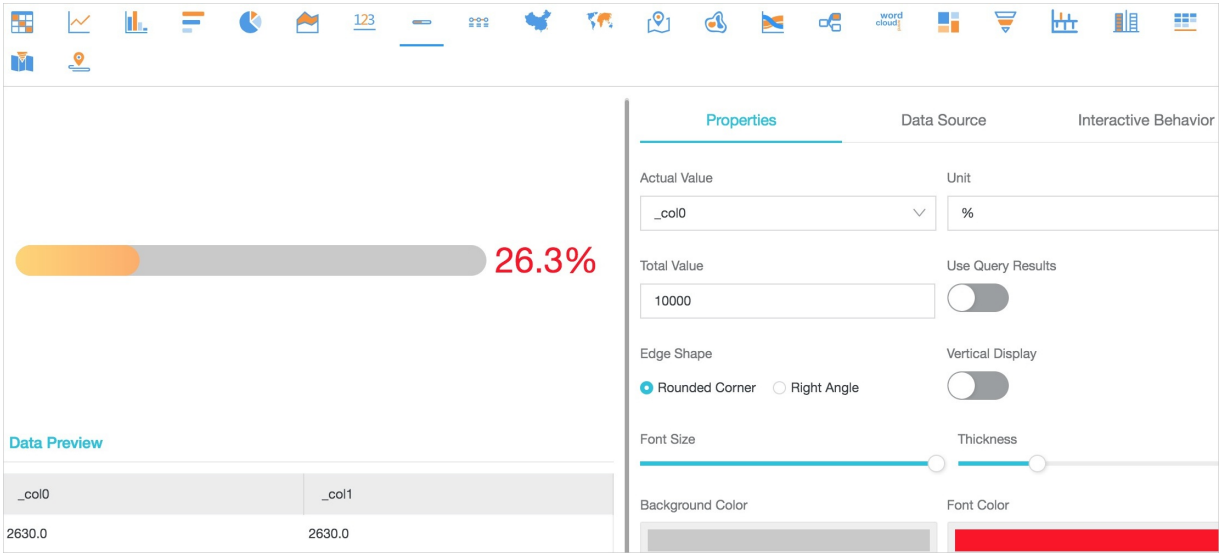
Parameter	Description
Actual Value	The actual value in the chart. By default, data in the first row of the specified column is displayed.
Unit	The unit of the value in the progress bar.
Total Value	The maximum value indicated by the progress bar. Default value: 100.

Parameter	Description
Maximum Value Column	The maximum value in the specified column. If you turn on the **Use Query Results** switch, **Total Value** is replaced by **Maximum Value Column**. Then, you can select the maximum value from the query results for this parameter.
Use Query Results	If you turn on the Use Query Results switch, Total Value is replaced by Maximum Value Column. Then, you can select the maximum value from the query results for this parameter.
Edge Shape	The edge shape of the progress bar.
Vertical Display	Specifies whether to display the progress bar in vertical display mode.
Font Size	The font size of the value in the progress bar.
Thickness	The thickness of the progress bar.
Background Color	The background color of the progress bar.
Font color	The font color of the value in the progress bar.
Default Color	The default color of the progress bar.
Color Display Mode	The display mode of the progress bar.
Start Color	The start color of the progress bar. This parameter is available if **Gradient** is selected for **Color Display Mode**.
End Color	The end color of the progress bar. This parameter is available if **Gradient** is selected for **Color Display Mode**.
Display Color	The display color of the progress bar. This parameter is available if **Display by Rule** is selected for **Color Display Mode**. **Note** The value of **Actual Value** is compared with that of **Threshold** based on the condition specified by **Operator**. If the actual value matches the condition specified by **Operator**, the progress bar is displayed in the color specified by Display Color. Otherwise, the progress bar is displayed in the default color.
Operator	The condition that is used to specify the color of the progress bar. This parameter is available if **Display by Rule** is selected for **Color Display Mode**.
Threshold	The threshold that is used to specify the color of the progress bar. This parameter is available if **Display by Rule** is selected for **Color Display Mode**.

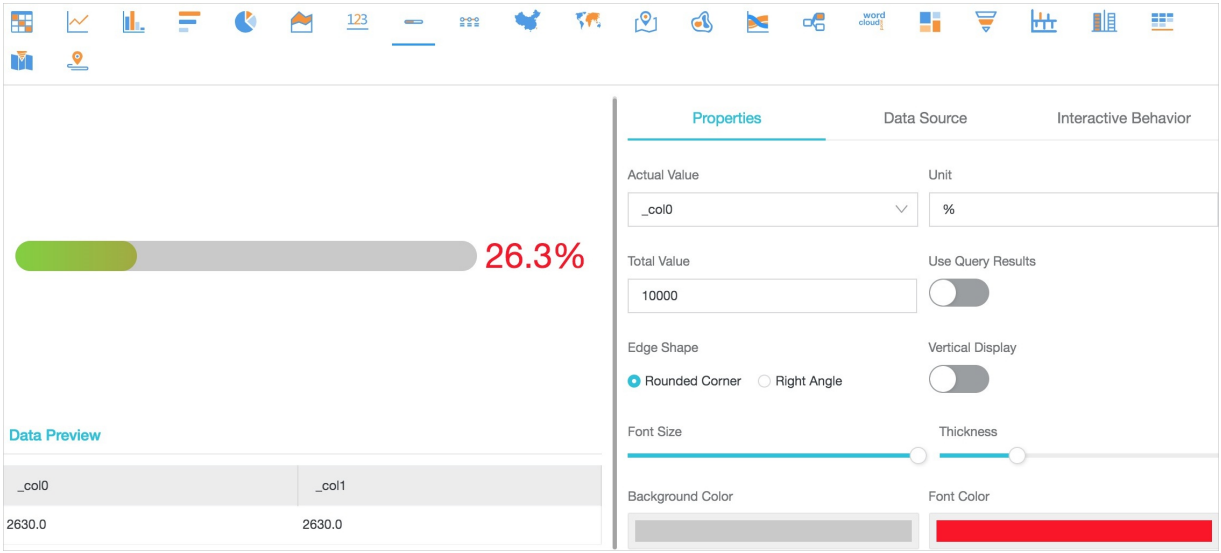
Examples

To view the percentage of a metric or the proportion of data, you can execute the following query statement:

```
* | select diff[1],diff[2] from (select compare( pv , 86400) as diff from (select count(1) as pv from log))
```



If you select **Display by Rule** for Color Display Mode, colors are dynamically displayed based on the specified rule. If the conditions of a rule are not matched, the default color is displayed.



28.1.4.11.1.10. Display query results on a map

You can color and mark a map to display geographic data. Log Service provides three types of maps: map of China, world map, and AMap. The display modes of an AMap include the anchor point and heat map. You can include specific functions in query statements to display analysis results as maps.

Components

- Map canvas
- Color area

Properties

Parameter	Description
Location information	The location information that is recorded in logs. The information is displayed in one of the following dimensions based on the map type: • Provinces (Map of China) • Country (World Map) • Longitude/Latitude (AMap)
Value Column	The data volume of the location information.

Procedure

1. On the Search & Analysis page, enter a query statement in the search box, specify a time range, and then click **Search & Analyze**.
   - To display query results on a map of China, include the `ip_to_province` function in a query statement.
   - To display query results on a world map, include the `ip_to_country` function in a query statement.
   - To display query results on an AMap, include the `ip_to_geo` function in a query statement.
2. On the **Graph** tab, click the icon.
3. Configure the properties of the map.

Map of China

To display query results on a map of China, you can execute the following query statement that includes the `ip_to_province` function:

- SQL statement

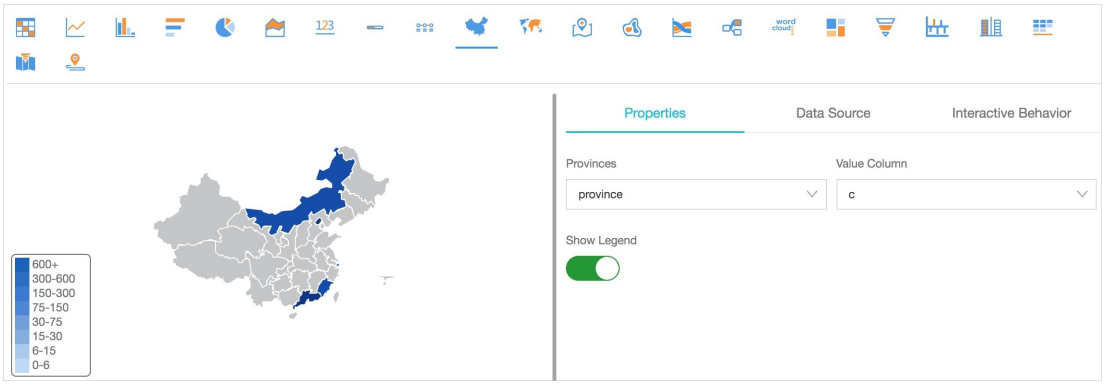
```
* | select  ip_to_province(remote_addr) as address, count(1) as count group by address order by co
unt desc limit 10
```

- Dataset

address	count
Guangdong	163
Zhejiang	110
Fujian	107
Beijing	89
Chongqing	28
Heilongjiang	19

Select address for *Provinces* and count for *Value Column*.

Map of China



World Map

To display query results in a world map, you can execute the following query statement that includes the `ip_to_country` function:

- SQL statement

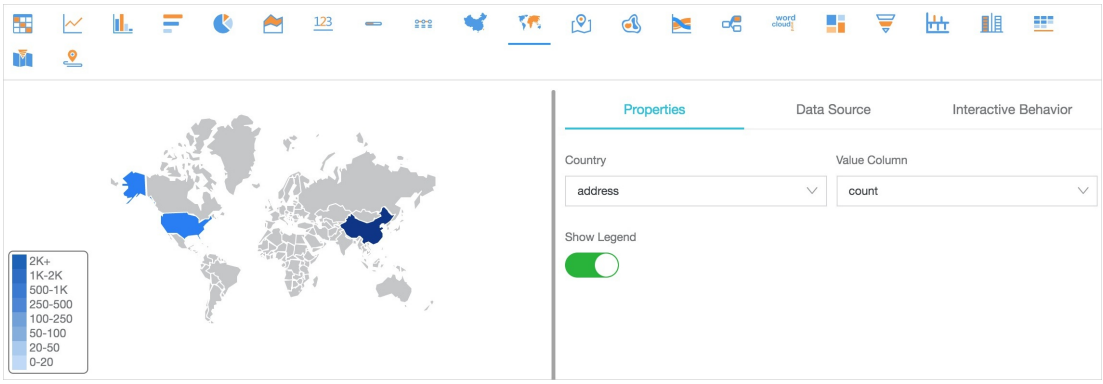
```
* | select  ip_to_country(remote_addr) as address, count(1) as count group by address order by cou
nt desc limit 10
```

- Dataset

address	count
China	8354
United States	142

Select address for *Country* and count for *Value Column*.

World Map



AMap

To display query results on an AMap, you can execute the following query statement that includes the `ip_to_geo` function. The address column in the dataset contains the latitude and longitude values, which are separated by a comma (,). If the longitude and latitude values are contained in two separate columns named lng and lat, you can use the `concat('lat', ',', lng')` function to combine the two columns into one column.

- SQL statement

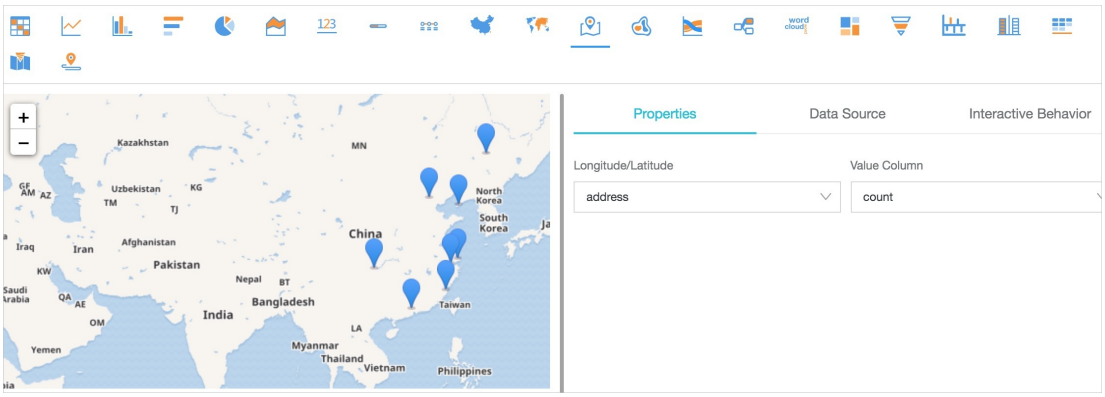
```
* | select  ip_to_geo(remote_addr) as address, count(1) as count group by address order by count d
esc limit 10
```

- Dataset

address	count
39.9289,116.388	771
39.1422,117.177	724
29.5628,106.553	651
30.2936,120.161420	577
26.0614,119.306	545
34.2583,108.929	486

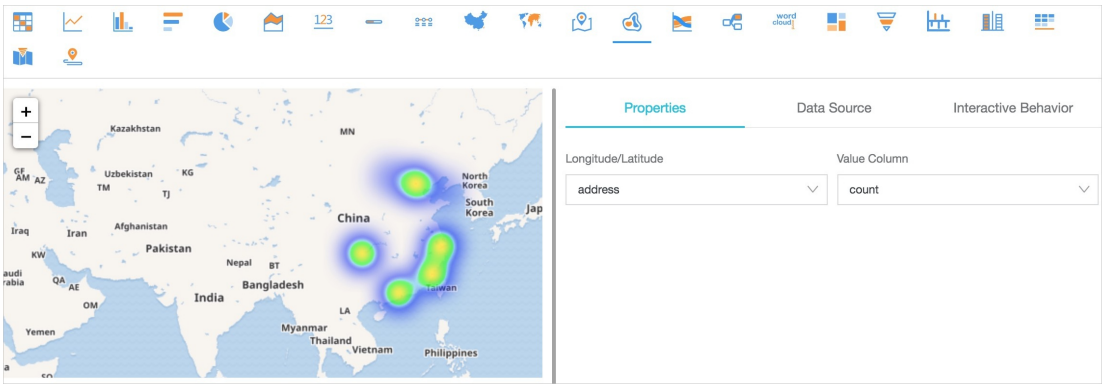
Select address for *Longitude/Latitude* and count for *Value Column*.

AMap: Anchor points



By default, the display mode of the anchor points is returned. If the anchor points are densely distributed on the map, you can switch the display mode to heat map.

AMap: Heat map



28.1.4.11.1.11. Flow chart

The flow chart, also known as ThemeRiver, is a stacked area chart around a central axis. The banded branches with different colors indicate different categorical data. The band width indicates the numeric value. The time information of the data is mapped to the X-axis by default. A flow chart can display the data of three parameters.

You can select the line chart or column chart for the Chart Types parameter. The column chart is stacked by default. Each category of data starts from the top of the last column of categorical data.

Components

- X-axis (horizontal)
- Y-axis (vertical)
- Band

Procedure

1.
2. On the **Graph** tab, click the icon.
3. On the **Properties** tab, configure the properties of the flow chart.

Properties

Parameter	Description
Chart Types	The type of the chart. Valid values: Line Chart, Area Chart, and Column Chart. Default value: Line Chart.
X Axis	The sequential data. In most cases, time series is selected.
Y Axis	The numeric data. You can configure one or more fields on the Y-axis.
Aggregate Column	The field information required to be aggregated as the third point for comparison.
Legend	The position where the legend is located in the chart. Valid values: Top, Bottom, Left, and Right.
Format	The format in which data is displayed.
Margin	The distance between an axis and the borders of the chart. Valid values: **Top Margin**, **Bottom Margin**, **Right Margin**, and **Left Margin**.

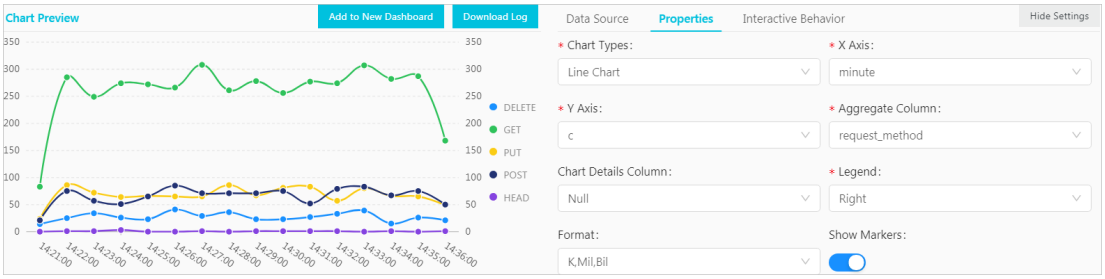
Examples

The flow chart is suitable to display data of three parameters, including the time, categories, and numeric values. In this example, you can execute the following query statement:

```
* | select date_format(from_unixtime(__time__ - __time__% 60), '%H:%i:%S') as minute, count(1) as c, request_method group by minute, request_method order by minute asc limit 100000
```

Select `minute` for the X-axis, `c` for the Y-axis, and `request_method` for Aggregate Column.

Flow chart



28.1.4.11.1.12. Display query results in a Sankey diagram

A Sankey diagram is a specific type of flow chart. It is used to describe the flow from one set of values to another set of values.

Sankey diagrams are applicable to scenarios such as network traffic flows. A Sankey diagram contains the values of three fields: `source`, `target`, and `value`. The `source` and `target` fields describe the source and target nodes and the `value` field describes the flows from the `source` node to the `target` node.

Features

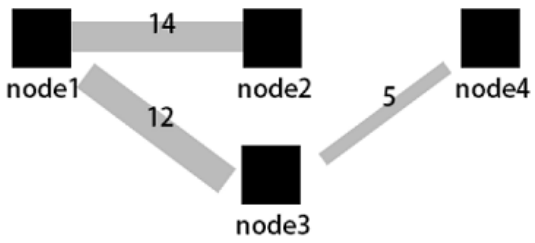
A Sankey diagram has the following features:

- The start flow is equal to the end flow. The overall width of all main edges is equal to the total sum of all branch edges. This allows you to manage and maintain a balanced flow of all traffic.
- The edge width in a row represents the traffic distribution in a specific status. The edge width in a row represents the distribution of traffic.
- The width of an edge between two nodes represents the flow volume of a status.

The following table lists the data that can be displayed in a Sankey diagram.

source	target	value
node1	node2	14
node1	node3	12
node3	node4	5
...	...	...

The following figure shows the data relationships in a Sankey diagram.



Components

- Node
- Edge

Procedure

1.
2. On the **Graph** tab, click the icon.
3. On the **Properties** tab, configure the properties of the Sankey diagram.

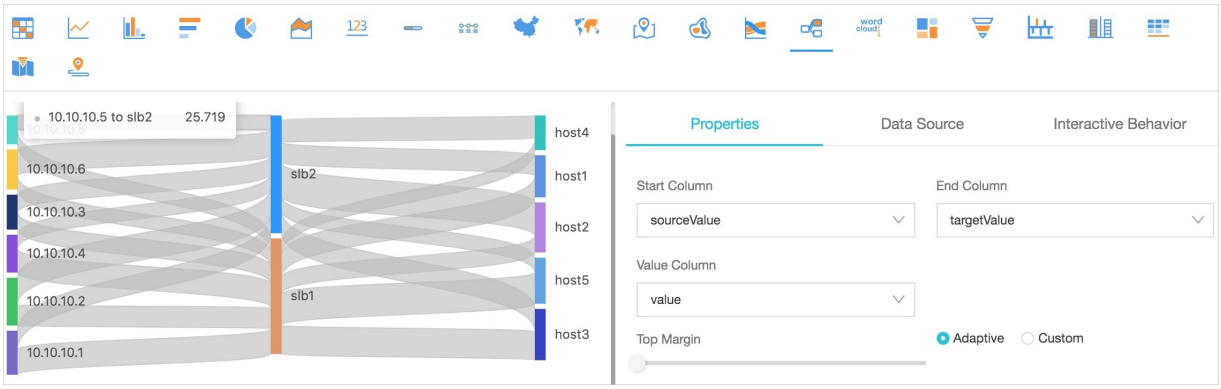
Properties

Parameter	Description
Start Column	The start node.
End Column	The end node.
Value Column	The value that indicates the flow volume from the start node to the end node.
Margin	The distance between an axis and the borders of the chart. Valid values: **Top Margin**, **Bottom Margin**, **Right Margin**, and **Left Margin**.

Example of a Sankey diagram

If a log entry contains the `source`, `target`, and `value` fields, you can use a Nested subqueries statement to obtain the sum of all `steamValue` values.

```
* | select sourceValue, targetValue, sum(streamValue) as streamValue from (select sourceValue, target
Value,
 streamValue, __time__ from log group by  sourceValue, targetValue, streamValue, __time__ order by __
time__ desc) group by sourceValue,
 targetValue
```



28.1.4.11.1.13. Display query results on a word cloud

A word cloud visualizes text data. It is a cloud-like and colored image that consists of words. You can use a word cloud to display a large amount of text data. The font size or color of a word indicates the significance of the word. This allows you to identify the most significant words in an efficient way.

Components

The words in a word cloud are sorted.

Procedure

1.
2. On the **Graph** tab, click the icon.
3. On the **Properties** tab, configure the properties of the word cloud.

Properties

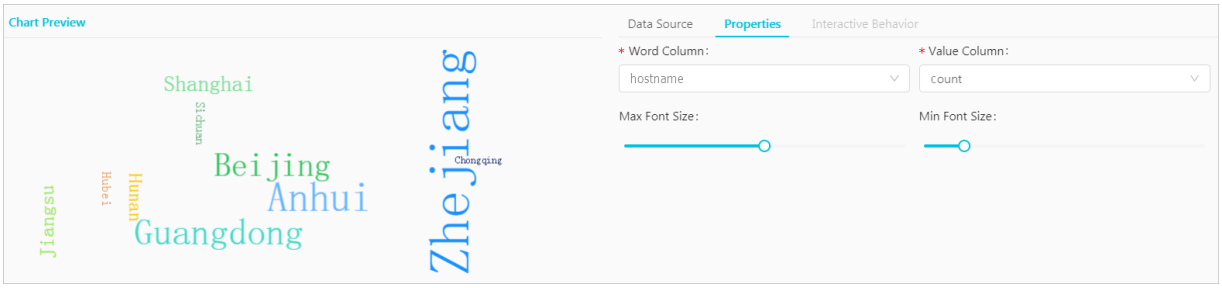
Parameter	Description
Word Column	The words to be displayed.
Value Column	The numeric value that corresponds to a word.
Font Size	The font size of a word. • The minimum font size ranges from 10 pixels to 24 pixels. • The maximum font size ranges from 50 pixels to 80 pixels.

Examples

To query the distribution of hostnames in NGINX logs, execute the following query statement:

```
* | select domain, count(1) as count group by domainorder by count desc limit 1000
```

Select `hostname` for Word Column and `count` for Value Column.



28.1.4.11.1.14. Display query results on a treemap chart

A treemap chart includes multiple rectangles that represent the data volume. A larger rectangle area represents a larger proportion of the categorical data.

Components

Sorted rectangles

Procedure

1.
2. On the **Graph** tab, click the icon.
3. On the **Properties** tab, configure the properties of the treemap chart.

Properties

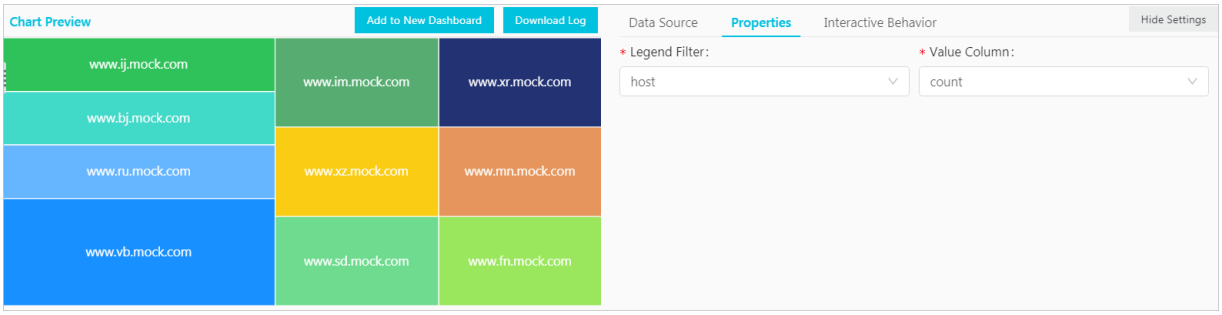
Parameter	Description
Legend Filter	The field that includes categorical data.
Value Column	The numeric value of a field. A greater field value represents a larger rectangle area.

Examples

To query the distribution of hostnames in NGINX logs, execute the following query statement:

```
* | select hostname, count(1) as count group by hostname order by count desc limit 1000
```

Select **hostname** for `Legend Filter` and **count** for `Value Column`.



28.1.4.11.2. Dashboard

28.1.4.11.2.1. Overview

A dashboard provided by Log Service is a platform where you can analyze data in real time. You can add multiple charts to a dashboard for data analysis. Each chart is a visualized search and analytic statement.

A dashboard allows you to view the charts of multiple search and analytic statements at one time. When you open or refresh the dashboard, the statements of the charts run automatically.

After you add a chart to a dashboard, you can configure Drill-down analysis for the chart. Then you can click the chart on the dashboard to further analyze data and obtain more fine-grained analysis results.

Limits

- You can create a maximum of 50 dashboards for a project.
- Each dashboard can contain a maximum of 50 analysis charts.

Features

A dashboard has two modes: display mode and edit mode.

- Configure the display mode of a dashboard

  In the display mode, you can configure multiple display settings on the dashboard page.

  - Dashboard: You can specify the time range, the automatic refresh interval, full screen, and the display mode of the title for the dashboard, configure alerts for all charts on the dashboard, and filter chart data based on the Configure and use a filter on a dashboard of a Logstore.
  - Chart: You can view the analysis details of a specified chart, specify the time range and configure alerts for the chart, download logs and the chart, and check whether drill-down analysis is configured for the chart.

- Edit mode

  In the edit mode, you can change the configurations of the dashboard and charts.

  - Dashboard: You can use a dashboard as a canvas and add Markdown chart, custom charts, text, icons, and other chart elements to the dashboard. You can also add lines between chart elements that are self-adaptive to the positions of the charts. You can also add Configure and use a filter on a dashboard of a Logstore, which can be used to filter chart data in the display mode. In addition, you can configure display gridlines to help arrange chart elements such as icons in an orderly manner.
  - Chart: You can also edit a chart on the dashboard. You can modify the statement, properties, and interactive behavior such as drill-down analysis of the chart.

28.1.4.11.2.2. Create and delete a dashboard

This topic describes how to create and delete a dashboard in a Logstore. In the Log Service console, you can run a query statement and visualize the query result in a chart. After you complete the configurations of the chart, you can add the chart to a dashboard. Each dashboard can display up to 50 charts, which support multiple formats and custom settings.

Prerequisites

The indexing feature of the Logstore is enabled and configured. For more information, see Enable the index feature and configure indexes for a Logstore.

Create a dashboard

1. Log on to the Log Service console.
2. Click a project name.
3. Click the icon next to the name of the Logstore, and then select **Search & Analysis**.
4. Enter a query statement in the search box, and then click **Search & Analyze**.
5. On the **Graph** tab that appears, configure the chart properties on the **Properties** tab.
6. (Optional)Set a placeholder variable.

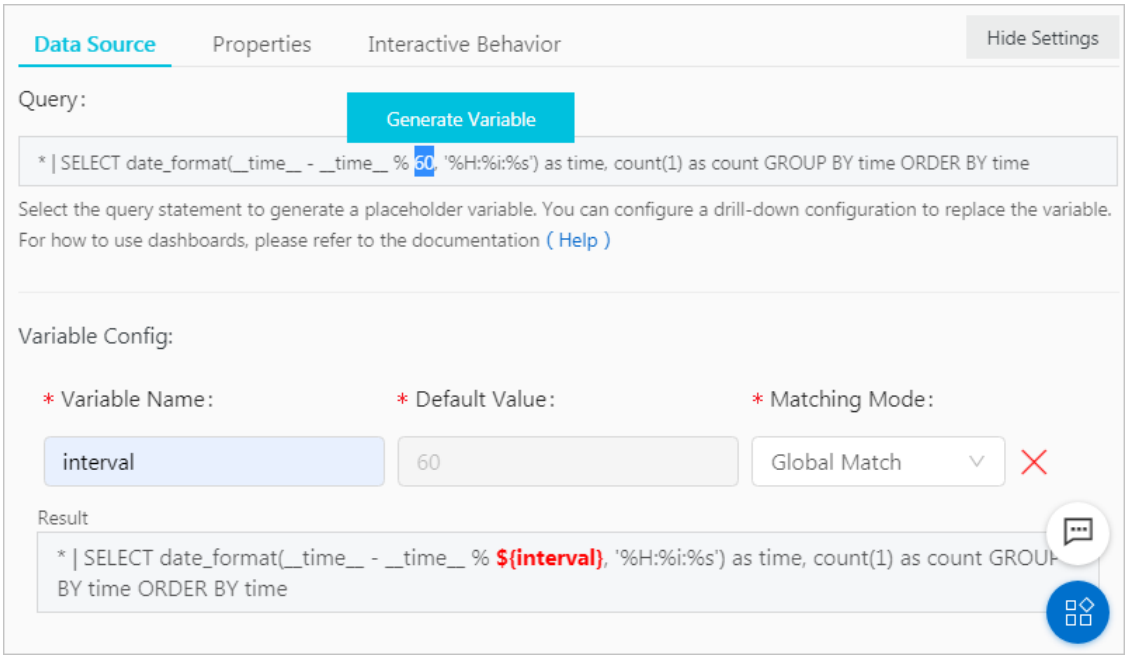
   For example, you have configured a drill-down event for another chart. This drill-down event redirects you to the current dashboard. You have also specified a placeholder variable for the query statement of the preceding chart. When you click a chart value to trigger the drill-down event, you are redirected to the current dashboard. The placeholder variable is replaced by the chart value and the current dashboard is refreshed by the new query statement. For more information, see Drill-down analysis.

   i. Click the **Data Source** tab, and then select a part of the query statement in the **Query** field.

   ii. Click **Generate Variable** to generate a placeholder variable.

iii. Set the parameters in the **Variable Config** section.

Parameter	Description
Variable Name	The name of the placeholder variable. If the name of the placeholder variable is the same as the variable specified in the chart, the placeholder variable is replaced with the chart value when the drill-down event is triggered.
Default Value	The default value of the placeholder variable in the current dashboard.
Matching mode	You can select Global Match or Exact Match.
Result	The query statement that contains the specified variable.



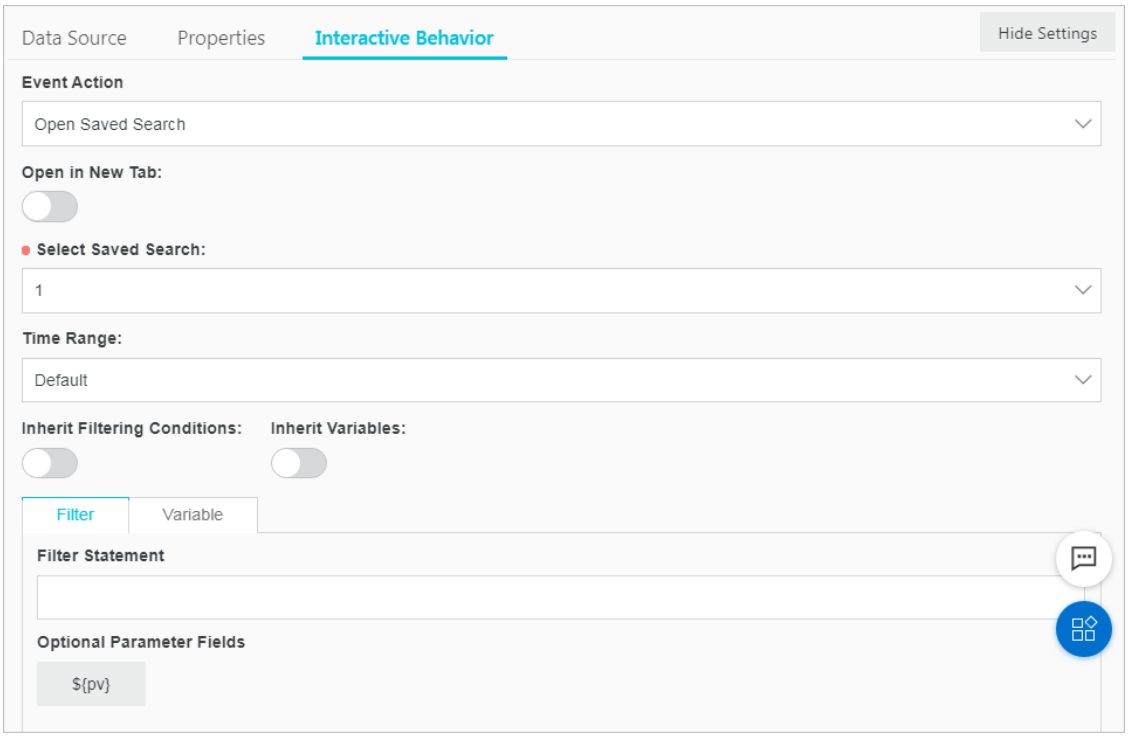
7. Configure a drill-down event.

   After you configure a drill-down event, you can click the chart on the dashboard for a more detailed analysis. For example, you can be redirected to another dashboard or a saved search. For more information, see Drill-down analysis.

   i. Click the **Interactive Behavior** tab.

   ii. Select an **Event Action**.

iii. Set the parameters of the selected event action.



8. Click **Add to New Dashboard** and specify the dashboard name and chart name.

Parameter	Description
Operation	◦ **Add to Existing Dashboard**: Add the chart to an existing dashboard. ◦ **Create Dashboard**: Create a dashboard and then add the chart to the dashboard.
Dashboards	Select an existing dashboard name. **Note** This parameter is required only when you set the **Operation** parameter to **Add to Existing Dashboard**.
Dashboard Name	Enter a dashboard name. **Note** This parameter is required only when you set the **Operation** parameter to **Create Dashboard**.
Chart Name	Enter a name for the current chart. The chart name is displayed as the chart title in the dashboard.

9. Click **OK**.

You can add multiple charts to a dashboard.

Delete a dashboard

You can delete a dashboard if you no longer need it. However, you cannot recover a deleted dashboard.

1. Log on to the Log Service console, and then click the destination project name.

2. In the left-side navigation pane, click the **Dashboard** icon.
3. Click the icon next to the dashboard that you want to delete, and then select **Delete**.

28.1.4.11.2.3. Configure the display mode of a dashboard

This topic describes how to configure the display mode of a dashboard. By default, you can view all charts in a dashboard in the display mode. When you configure the display mode, you can add chart elements, enable automatic refresh, and set the title display mode.

Set a query time range for a dashboard of a Logstore

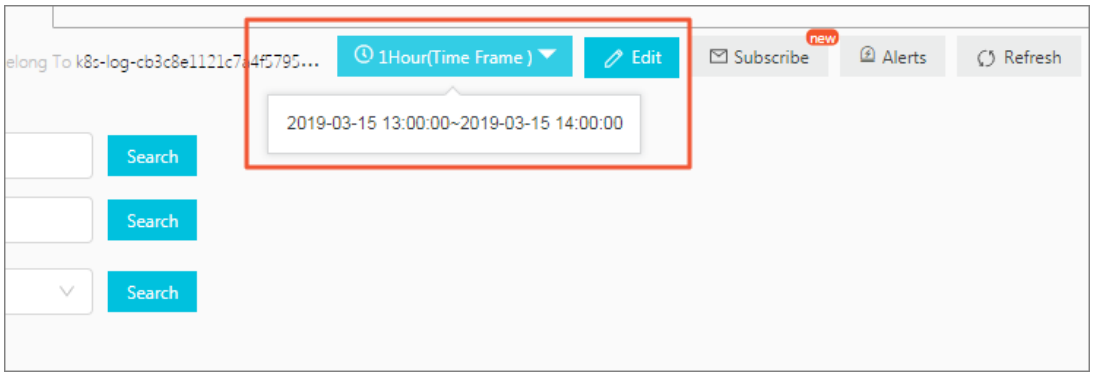
By default, all charts in a dashboard use the query time range that is set for the dashboard. For more information about how to set a query time range for a single chart, see Set a query time range for a chart.

> **Note** On the dashboard page, you can click **Time Range** to specify a time range for a query. The specified query time range is used only for the current query. The next time you open the dashboard, the system will display the analysis results in the default query time range.

1. Log on to the Log Service console.
2. Click a project name.
3. In the left-side navigation pane, click the **Dashboard** icon.
4. Click the icon next to the dashboard, and then select **Details** from the drop-down list.
5. Click **Time Range** to set a time range.

   You can set one of the following time ranges:

   - Relative: queries log data obtained in a time range of 1 minute, 5 minutes, 15 minutes, or other time ranges that end with the current time, accurate to the second. For example, if the current time is 19:20:31 and you select 1Hour as the relative time, the charts on the dashboard display the analysis results of the log data queried from 18:20:31 to 19:20:31.
   - Time Frame: If you select or customize a time range less than one hour (for example, 1 minute, 5 minutes, and 15 minutes), log data obtained in the time range that ends with the current time is queried, accurate to the minute. If you select or customize a time range greater than one hour, log data obtained on the hour before the current time is queried. For example, if the current time is 19:20:31 and you select 1Hour as the time frame, the charts on the dashboard display the analysis results of the log data queried from 18:00:00 to 19:00:00.
   - Custom: queries log data obtained in a specified time range.
6. Move the pointer over the **Time Range** button to confirm the specified time range.



Switch to the edit mode

Click **Edit** to switch to the edit mode of the dashboard. In the edit mode, you can add Markdown chart, custom charts, text, icons, and other chart elements to the dashboard. For more information, see Edit mode.

Set alerts

On the dashboard page, choose **Alerts > Create** to create an alert. Choose **Alerts > Modify** to modify an alert. An alert must be associated with one or more charts.

For more information, see Configure alerts.

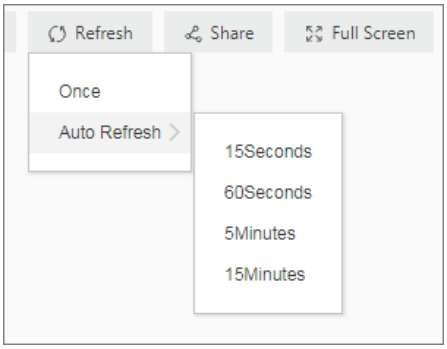
Set a refresh method

You can manually refresh the dashboard, or set an automatic refresh interval for the dashboard.

- To manually refresh the dashboard, choose **Refresh > Once**.
- To set an automatic refresh interval for the dashboard, choose **Refresh > Auto Refresh**, and then select an interval.

  The **Auto Refresh** interval can be 15 seconds, 60 seconds, 5 minutes, or 15 minutes.

> **Note** If your browser is inactive, the automatic refresh interval may be inaccurate.



Share a dashboard

To share a dashboard with authorized users, click **Share** to copy the link of the dashboard page and then send the link to the users. The shared dashboard page uses the settings of the dashboard at the time of sharing. The settings include the time range of charts and chart title format.

> **Note** Before you share the dashboard with other users, you must grant relevant permissions to them.

Display a dashboard in full screen

Click **Full Screen**. Then the charts on a dashboard are displayed in full screen.

Set the chart title format

On the dashboard page, click **Title Configuration**. Available title formats include:

- Single-line Title and Time Display
- Title Only
- Time Only

Reset the query time range

To restore the default query time ranges of all charts on the dashboard, click **Reset Time**.

Select chart view

- View analysis details of a chart

To view analysis details of a chart, move the pointer over the More icon in the upper-right corner of the chart, and then select **View Analysis Details**. The corresponding Search & Analysis page appears, showing the query statement and property settings.

- Set the query time range for a chart

  To set the query time range for a chart, move the pointer over the More icon in the upper-right corner of the chart, and then select **Select Time Range**. The settings are valid only for the current chart.

- Set an alert for a chart

  To set an alert for a chart, move the pointer over the More icon in the upper-right corner of the chart, and then select **Create Alert**. For more information, see Configure alerts.

- Download log analysis results of a chart

  To download analysis results of a chart, move the pointer over the More icon in the upper-right corner of the chart, and then select **Download Log**.

- Download a chart

  To download a chart, move the pointer over the More icon in the upper-right corner of the chart, and then select **Download Chart**.

- Check whether a drill-down event is configured for a chart

  To check whether a drill-down event is configured for a chart, move the pointer over the More icon in the upper-right corner of the chart. Then, check the color of the hand icon at the bottom of the shortcut menu. If the icon is red, a drill-down event is configured for the chart. If the icon is gray, no drill-down event is configured for the chart.

> **Note** Different charts in a dashboard have different shortcut menus. For example, you cannot use the shortcut menu of a custom chart or Markdown chart to view analysis details because they are not analysis charts.

28.1.4.11.2.4. Edit mode

You can click the Edit button on the dashboard page to enter the edit mode. Then you can change the configurations of the dashboard and charts on the dashboard.

- Dashboard:
  - You can modify the dashboard name in the upper-left corner of the page.
  - You can use a dashboard as a canvas and add Markdown chart, custom charts, text, icons, and other chart elements to the dashboard.
  - You can add lines between chart elements. The lines are adaptive to the positions of the charts.
  - You can add a filter to the dashboard. Then you can filter chart data in the display mode. For more information, see Configure and use a filter on a dashboard of a Logstore.
  - In addition, you can configure display grid lines to arrange chart elements such as icons in order.
  - You can use the menu bar to manage the chart property settings on the dashboard. For example, you can perform the add, delete, and cancel operations on a chart. You can also configure the size and location of a chart and move a chart to the top or bottom of the dashboard.
- Chart: You can edit a chart on the dashboard. For example, you can modify the statement, properties, and interactive behavior such as drill-down analysis of a chart.

> **Note** All changes to the dashboard take effect only after you click **Save** in the upper-right corner of the page.

Chart elements

In the edit mode of a dashboard, you can add the following chart elements:

- Common icons

  Log Service allows you to display common icons on a dashboard page. You can drag an icon from the menu bar and drop the icon to a specified position.

- Text

  You can drag the text icon from the menu bar and drop the icon to a specified position. Double-click the text box to modify the text.

- Markdown chart

  You can add a Markdown chart to a dashboard and edit the chart with the Markdown syntax.

  Drag the Markdown icon from the menu bar and drop the icon to a specified position. Select **Edit** from the More icon. Then you can set the Markdown content.

- Filter

  You can add a filter to a dashboard. Then you can add search conditions or replace placeholder variables in query statements.

  Click the filter icon in the dashboard menu bar. On the page that appears, configure the filter based on your needs. By default, the filter is in the upper-left corner of the dashboard page. To modify the settings of a filter, you can select **Edit** on the More icon in the upper-right corner of the page.

- Customize SVG

  You can upload a Scalable Vector Graphics (SVG) file to a dashboard. Click the SVG icon in the menu bar. On the page that appears, click the box or drag an SVG file to the box to upload the file.

  > **Note** The size of an SVG file cannot exceed 10 KB.

- Customize image's HTTP link

  You can upload the HTTP link of an image to a dashboard. Click the Customize image's HTTP link icon in the menu bar. On the page that appears, enter the HTTP link of an image and click **OK**.

Layout

In the edit mode, all charts and chart elements are placed on a canvas. You can drag and scale each chart, except for the lines. The horizontal width of a chart is up to 1,000 units. Each unit is equal to `current browser width/1,000`. The vertical height is unlimited and each unit is equal to 1 pixel. Before you arrange a chart on the dashboard, you can click **Display gridlines** to better arrange the position of the chart and the spacing between charts.

You can perform the following operations to arrange a chart on the dashboard:

- Adjust the position of a chart
  - You can drag a chart and drop the chart to a specified position.
  - After you specify a chart, you can click **L** and **T** to adjust the chart position.
- Adjust the width and height of a chart
  - After you specify a chart, you can drag the chart in the lower-right corner to resize the chart.
  - After you specify a chart, you can also specify the **W** and **H** parameters to resize the chart.
- Add lines to connect charts

  You can add a directional line between two charts. When you adjust the position and size of the charts, the line automatically moves to display the relative position between the two charts.

After you select a chart, four squares appear on the border of the chart. These squares indicate the starting point of the connection line. Press and hold one square, and the area where the end point of the connection line is automatically displayed. Move the pointer to this position to connect the two charts.

- You can move a chart to the top or bottom of the dashboard. After you select a chart, you can use the menu bar to move the chart to the top or bottom.

Chart configurations

In the edit mode of a dashboard, you can perform the following operations on chart elements:

- Edit: modifies the query statement, properties, and interactive behavior such as drill-down analysis of a chart.
  i. In the upper-right corner of the dashboard page, click **Edit**.
  ii. Find the target chart and choose > **Edit**.
  iii. Modify the query statement of the chart, **Properties**, **Data Source**, or **Interactive Behavior**.
  iv. Click **Preview**, and then click **OK**.
  v. In the upper-right corner of the dashboard page, click **Save**.
- Copy: creates a copy of the specified chart and retains all configurations.
  i. In the upper-right corner of the dashboard page, click **Edit**.
  ii. Find the target chart and choose > **Copy**.
  iii. Drag the chart copy and drop it to a specified position, and then set the top margin, left margin, and size of the copy.
  iv. In the upper-right corner of the dashboard page, click **Save**.
- Delete: deletes the specified chart from the dashboard.
  i. In the upper-right corner of the dashboard page, click **Edit**.
  ii. Find the target chart and choose > **Delete**.
  iii. In the upper-right corner of the dashboard page, click **Save**.

28.1.4.11.2.5. Drill-down analysis

This topic describes how to configure drill-down analysis for a chart of a Logstore. When you add a chart to a dashboard, you can modify the configurations in the drill-down list to obtain deeper data analysis results.

Prerequisites

- The index feature of the Logstore is enabled and configured. For more information, see Enable the index feature and configure indexes for a Logstore.
- A saved search, a dashboard, and a custom link to which you want to be redirected are configured.
- A placeholder variable is specified in the saved search or the query statement of a chart added to a dashboard to which you are redirected if you want to add a variable. For more information, see Saved search and Create and delete a dashboard.

Context

Drilling is essential for data analysis. This feature allows you to analyze data in a fine-grained or coarse-grained way. This feature includes drill-up and drill-down analysis. You can implement drill-down analysis to gain a deeper insight into data and make a better decision.

Log Service supports drill-down analysis of the following charts: tables, line charts, column charts, bar charts, pie charts, individual value plots, area charts, and treemap charts.

Procedure

1. Log on to the Log Service console.
2. Click the target project name.
3. Click the icon next to the name of the Logstore, and then select **Search & Analysis** from the drop-down list.
4. Enter a query statement in the search box, set a time range, and then click **Search & Analyze**.
5. On the **Graph** tab that appears, select a **chart type**, and then configure the parameters on the **Properties** tab of the chart.
6. Click the **Interactive Behavior** tab. On this tab, configure the **Event Action** for drill-down analysis.
   - **Disable**: disables drill-down analysis.
   - **Open Logstore**: sets the drill-down event to open a Logstore.

     If you configure a filter statement on the Interactive Behavior tab, the filter statement automatically fills the Search & Analyze search box of the redirected Logstore page when you click a value on the chart.

Parameter	Description
Select Logstore	The name of the Logstore to which you want to be redirected.
Open in New Tab	If you turn on this switch, the specified Logstore is opened on a new tab when the interactive behavior is triggered.
Time Range	The query time range of the Logstore to which you are redirected. Valid values: ■ **Default**: The default time range (15 minutes, accurate to the second) is used for a query statement of the redirected Logstore. ■ **Inherit table time**: The time range that a statement queries in the redirected Logstore is the time range specified for the chart when the event is triggered. ■ Relative: The time range that a statement queries in the redirected Logstore is accurate to the second. ■ Time Frame: The time range that a statement queries in the redirected Logstore is accurate to the minute, hour, or day. Default value: **Default**.
Inherit Filtering Conditions	If you turn on the **Inherit Filtering Conditions** switch, the filtering conditions added to the dashboard are synchronized to a query statement of the specified Logstore. The filtering conditions are added before the query statement by using the logical `AND` operator.
Filter	Enter a **Filter Statement** on the **Filter** tab. The filter statement can contain the **Optional Parameter Fields**. If you configure a filter statement on the **Filter** tab, the **filter statement** automatically fills the Search & Analyze search box of the redirected Logstore page when you click a chart value on the dashboard.

   - **Open Saved Search**: sets the drill-down event to execute a saved search.

     You can configure variables and a filter statement for a chart at the same time. When you click a value on the chart:

     ■ If you configure a variable for the chart, the variable value that you click replaces the placeholder variable that you have configured for the saved search and the saved search is executed for drill-down analysis.

- If you configure a filter statement, the filter statement is added to the saved search to which you are redirected.

Parameter	Description
Select Saved Search	The name of the saved search to which you want to be redirected. For more information about how to configure a saved search, see Saved search.
Open in New Tab	If you turn on this switch, the specified saved search is opened on a new tab when the interactive behavior is triggered.
Time Range	The time range of the saved search to which you want to be redirected. Valid values: ■ **Default**: The default time range (15 minutes, accurate to the second) is used for the saved search to which you are redirected. ■ **Inherit table time**: The time range of the saved search is the query time range of the chart that you configure on the dashboard. ■ Relative: The time range of the saved search is accurate to the second. ■ Time Frame: The time range of the saved search is accurate to the minute, hour, or day. Default value: **Default**.
Inherit Filtering Conditions	If you turn on the **Inherit Filtering Conditions** switch, the filtering conditions added on the dashboard are synchronized to the saved search of the specified Logstore. The filtering conditions are added before the saved search by using the logical `AND` operator.
Filter	Click the **Filter** tab, and then enter a **Filter Statement**. The filter statement can contain the **Optional Parameter Fields**. If you configure a filter statement on the **Filter** tab, the **Filter Statement** is added to the saved search when you click a chart value on the dashboard.
Variable	Click the **Variable** tab, click **Add Variable**, and then set the following parameters: ■ **Replace Variable Name**: the variable that triggers the drill-down event. When you click this variable, you are redirected to the specified saved search. ■ **Replace the value in the column**: the column where the value that you want to replace data with is located. To process multiple columns, you can specify the current column and other columns. The current column is the column on which you want to perform drill-down analysis. Specify the current column in the **Replace the value in the column** field. Other columns can be the fields in the chart for which you configure drill-down analysis. If the variable name of the saved search is the same as the added variable, the variable of the saved search is replaced with the chart value that triggers the drill-down event. This helps you use the saved search for analysis in an efficient way. **Note** ■ If you add a variable for drill-down analysis, you must first configure a placeholder variable for the saved search to which you want to be redirected. ■ You can add up to five placeholder variables for a saved search.

- **Open Dashboard**: sets the drill-down event to open a dashboard.

Analysis charts on a dashboard are visualized query results. When you click a chart value on the source dashboard:

- If you configure a variable for the chart and set a placeholder variable for the chart on the destination dashboard to which you want to be redirected, the placeholder variable is replaced with the chart value that you click.
- If you configure a filter statement, the filter statement is added to the query statement of the chart on the destination dashboard.

Parameter	Description
Select Dashboard	The name of the dashboard to which you want to be redirected. For more information, see Create and delete a dashboard.
Open in New Tab	If you turn on this switch, the specified dashboard is opened on a new tab when interactive behavior is triggered.
Time Range	Set the time range for the dashboard to which you want to be redirected. Valid values: ■ **Default**: After you are redirected to the dashboard by clicking a chart value on the current dashboard, the selected dashboard uses its original time range. ■ **Inherit table time**: After you are redirected to the selected dashboard, the time range of the chart on the selected dashboard is the time range of the chart specified on the source dashboard when the event is triggered. ■ Relative: After you are redirected to the selected dashboard, set the time range of the selected dashboard to the specified relative time. ■ Time Frame: After you are redirected to the selected dashboard set the time range of the selected dashboard to the specified time frame. Default value: **Default**.
Inherit Filtering Conditions	If you turn on the **Inherit Filtering Conditions** switch, the filtering conditions added on the current dashboard are synchronized to the dashboard to which you are redirected. The filtering conditions are added before the query statement by using the logical `AND` operator.
Filter	Click the **Filter** tab, and then enter a **Filter Statement**. The filter statement can contain the **Optional Parameter Fields**. If you have set the **Filter**, a filtering condition is added to the selected dashboard when you click a chart value on the current dashboard. The filtering condition is the specified **Filter Statement**.

Parameter	Description
Variable	Click the **Variable** tab, click **Add Variable**, and then set the following parameters: ■ **Replace Variable Name**: the variable that triggers drill-down analysis. When you click this variable, you are redirected to the selected dashboard. ■ **Replace the value in the column**: the column where the value that you want to replace data with is located. To process multiple columns, you can specify the **Default Column** and other columns. The **Default Column** is the current column on which you want to perform drill-down analysis. Other columns can be fields in the chart for which you configure drill-down analysis. If the variable in the query statement of the chart on the selected dashboard is the same as the added variable, the variable is replaced with the chart value that triggers the drill-down event. This changes the query statement of the chart on the selected dashboard. **Note** ■ If you add a variable for drill-down analysis, you must first configure a placeholder in the query statement for the selected dashboard to which you want to be redirected. ■ You can add up to five variables.

- **Custom HTTP Link**: sets the drill-down event to open a custom HTTP link.

  The path in the HTTP link is the path of the destination file in the server. When you add optional parameter fields to the path and click the chart value on the dashboard, the added parameter fields are replaced with the chart value. At the same time, you are redirected to the new HTTP link.

Parameter	Description
Enter Link	The destination address to which you want to be redirected.
Optional Parameter Fields	By clicking an optional parameter variable, you can replace a part of the HTTP link with the chart value that triggers a drill-down event.

7. Click **Add to New Dashboard** and set the dashboard name and chart name.

Example

You want to store collected NGINX access logs in a Logstore named accesslog, display the relevant analysis results on a dashboard named RequestMethod, and display the page view (PV) changes over time on a dashboard named destination_drilldown. You can configure drill-down analysis for the table of request methods, add the table to the RequestMethod dashboard, and then configure the drill-down event that redirects you to the destination_drilldown dashboard. When you click a request method on the table on the RequestMethod dashboard, you are redirected to the destination_drilldown dashboard. Then you can view the PV changes on the dashboard.

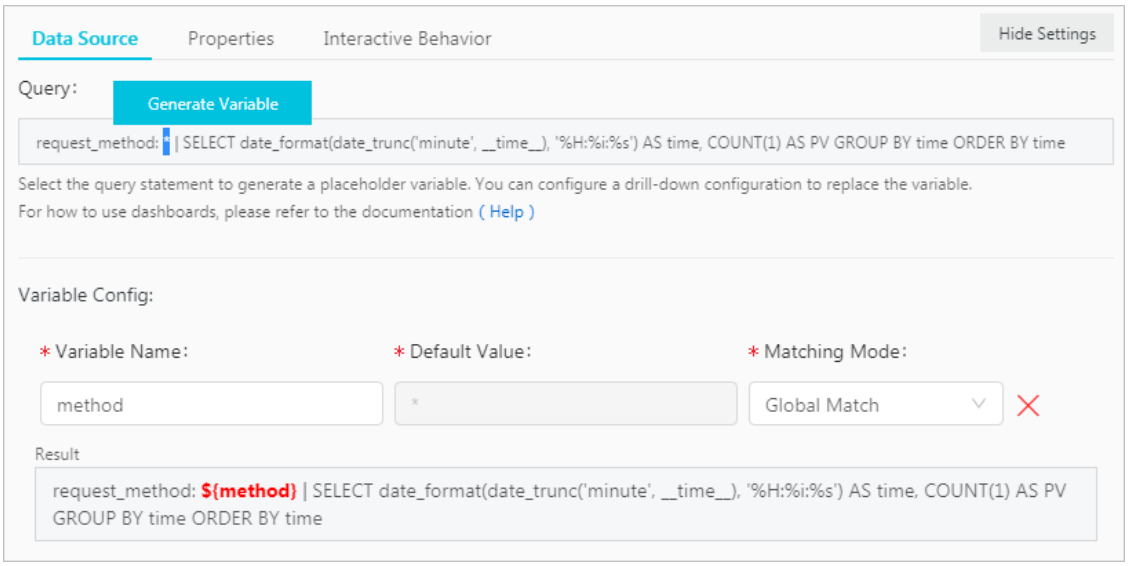
1. Set the dashboard to which you want to be redirected.
   i. Filter log data by request method and analyze how the PV of each request method changes over time.

      The query statement is shown as follows:

```
request_method: * | SELECT date_format(date_trunc('minute', __time__), '%H:%i:%s') AS time, C
OUNT(1) AS PV GROUP BY time ORDER BY time
```

ii. Use a line chart to display the query result and add the line chart to the dashboard named destination_drilldown.

Before you add the line chart to the dashboard, specify the asterisk (`*`) to generate a placeholder variable and set the variable name to method. If the variable name of another chart for which you configure drill-down analysis is method, when you click the variable value on the chart to trigger drill-down analysis, the asterisk (`*`) is replaced with the value that you click and the query statement of the line chart is performed.



2. Configure drill-down analysis for a chart and add the chart to the dashboard.

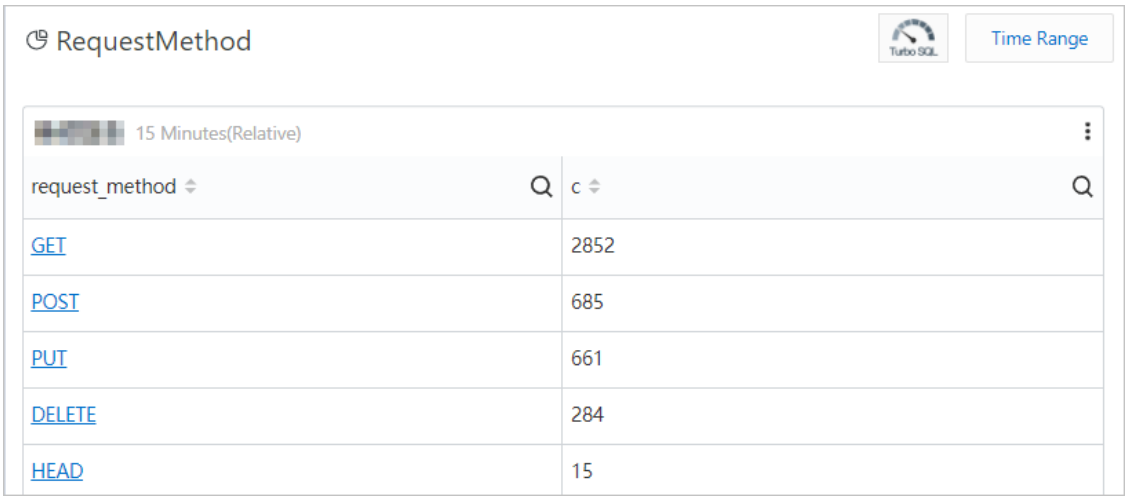
i. On the Search & Analysis page, enter a SQL statement to query the number of NGINX access log entries of each request method, and display the result in a table.

The query statement is shown as follows:

```
*|SELECT request_method, COUNT(1) AS c GROUP BY request_method ORDER BY c DESC LIMIT 10
```

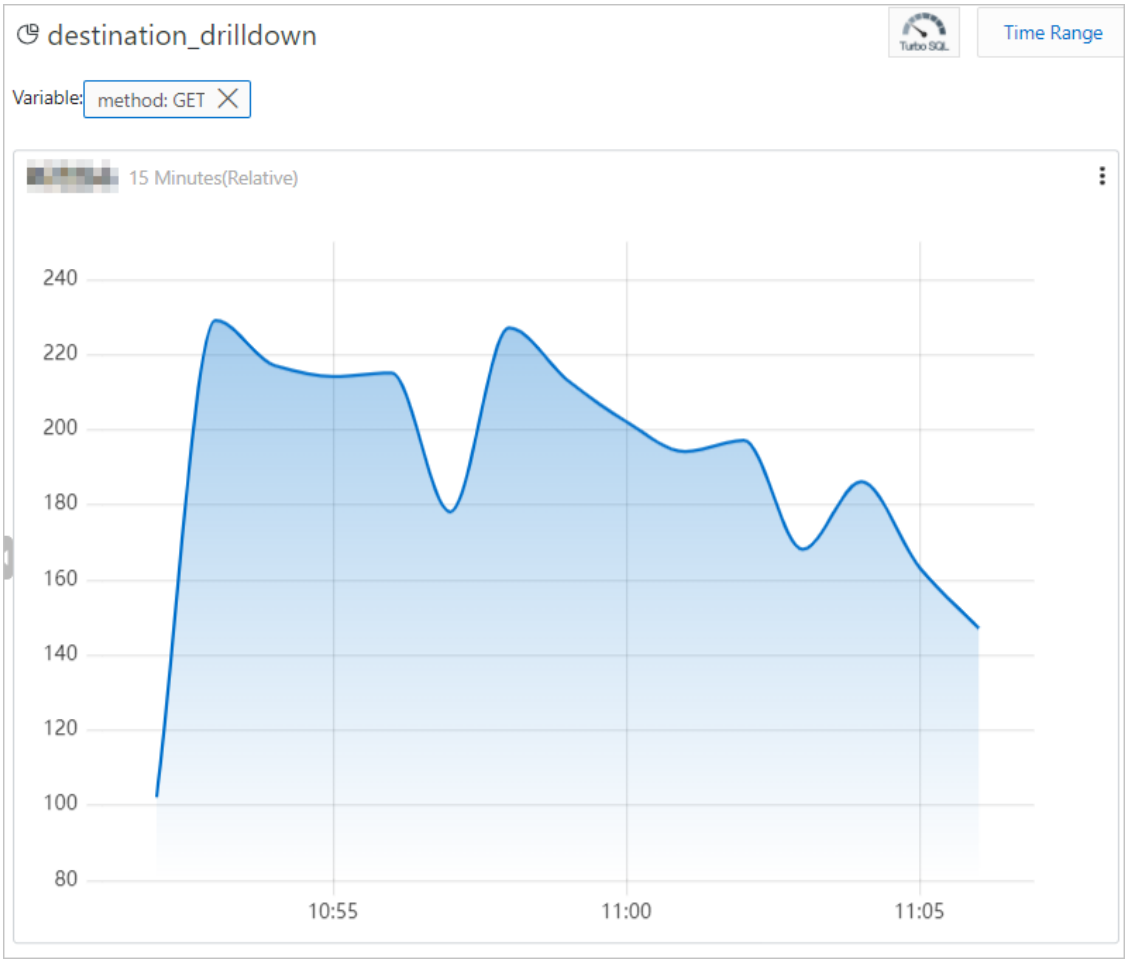
ii. Configure drill-down analysis for the `request_method` column in the table.

iii. Click the GET request on the RequestMethod dashboard.



iv. Redirected to the destination_drilldown dashboard.

You are redirected to the dashboard configured in step . The asterisk (`*`) in the query statement is replaced with `GET` . The dashboard shows how the PV of the GET request changes over time.



28.1.4.11.2.6. Configure and use a filter on a dashboard of a Logstore

This topic describes how to configure and use a filter on a dashboard of a Logstore. Filters help you refine search results or replace placeholder variables with specified values.

Prerequisites

- The index feature of the Logstore is enabled and configured. For more information, see Enable the index feature and configure indexes for a Logstore.
- A dashboard is created. For more information, see Create and delete a dashboard. A placeholder variable is specified for charts on the dashboard if the filter type is set to Replace Variable.

Context

Each chart on a dashboard is a visualized query statement. When you configure a filter on a dashboard, the specified filtering condition or variables apply to all charts on the dashboard. You can configure the following two types of filters:

- Filter: Add key-value pairs as a filtering condition before the query statement `[search query]` . The new query statement is `key:value AND [search query]` , which means to search the result of the original query

statement for log entries that contain `key:value`. For the **Filter** type, you can select or enter multiple key-value pairs. When you select multiple key-value pairs as filtering conditions, the logical `OR` operator is used between the pairs.

- Replace Variable: Specify a variable. If the dashboard contains a chart configured with the same placeholder variable, the placeholder variable in the query statement of the chart is replaced with the selected value.

Components

Each filter chart has one or more filters. Each filter consists of the following two components:

- The key, which indicates a filter operation.
- The values of the key.

Procedure

1. Log on to the Log Service console.
2. Click a project name.
3. In the left-side navigation pane, click the **Dashboard** icon.
4. In the dashboard list, click the name of the target dashboard.
5. On the dashboard page, click **Edit** to enter the edit mode.
6. Click the icon, and then set the filter parameters. Click **OK**.

Parameters of a filter

Parameter	Description
Filter Name	The name of the filter.
Display Settings	Valid values: ○ Title: specifies to add a title for a filter. You can turn on the Title switch to add a title for a filter. ○ Border: specifies to add borders for a filter. You can turn on the Border switch to add borders for a filter. ○ Background: specifies to add a white background for a filter. You can turn on the Background switch to add a white background for a filter.
Type	The filter type. Valid values: ○ If you select **Filter**, a List Item is a value of a key that is used to filter the results of a query statement. You can set multiple values for a key. After the filter takes effect, you can select one or more values on the dashboard to filter query results based on your needs. ○ If you select **Replace Variable**, a List Item is the value that replaces a specified variable. You can set multiple values for a variable. After the filter takes effect, you can select one or more values on the dashboard to filter query results based on your needs.

Parameter	Description
Key	◦ For the **Filter** type, the **Key** parameter specifies the key in the filtering condition. ◦ For the **Replace Variable** type, the **Key** parameter specifies the variable. **Note** The variable must be the same as the placeholder variable that you specified for charts. Otherwise, the placeholder variable cannot be replaced.
Alias	The alias of the key. This parameter is available only when you select **Filter**. After you set an alias for a key, the alias is displayed in the filter on the dashboard.
Global filter	This switch indicates whether to filter the specified values in all fields. This switch is turned off by default. The switch is available only when you select the **Filter** type. ◦ To filter the specified values in all fields, turn on the **Global filter** switch. ◦ To filter the specified values in specified keys, turn off the **Global filter** switch.
Static List Items	The values of the key that is used as a filtering condition. Click the **plus sign** and enter a value for the key in the text box.
Add Dynamic List Item	The dynamic value of the key retrieved by running the specified query statement. Turn on the **Add Dynamic List Item** switch, select a Logstore, and turn on the Inherit Filtering Conditions switch (specifies whether to include the filtering condition in the query statement). Enter a query statement in the search box, specify a time range, and then click **Search** to preview the dynamic values.

Examples

You can use a filter to modify the query statements of charts on a dashboard and replace placeholder variables in the charts on the dashboard. Each chart represents a query statement in the format of `[search query] | [sql query]`. The filter query is appended to the original query statement to filter data.

- If the filter type is Filter, the key-value pairs to be filtered are added before `[search query]` to form a new query statement by using the logical `AND` operator. The new query statement is `key:value AND [search query]`.
- If the filter type is Replace Variable, the filter queries all charts that contain the specified placeholder variables on the dashboard and replaces the placeholder variables with the selected `values`.

Example 1: Use different time granularities to analyze logs

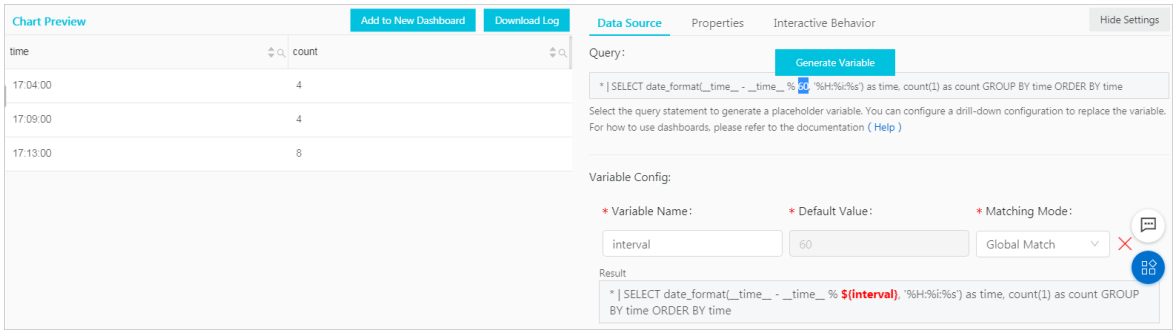
After you collect NGINX access logs, you need to query and analyze these logs in real time.

You can use a query statement to view the number of page views (PVs) per minute. However, if you want to view the number of PVs per second, you must modify the value of `__time__ - __time__ % 60` in the query statement. To simplify this operation, you can use a filter to replace variables in the query statement.

1. Use the following query statement to view the number of PVs per minute:

```
* | SELECT date_format(__time__ - __time__ % 60, '%H:%i:%s') as time, count(1) as count GROUP BY time ORDER BY time
```

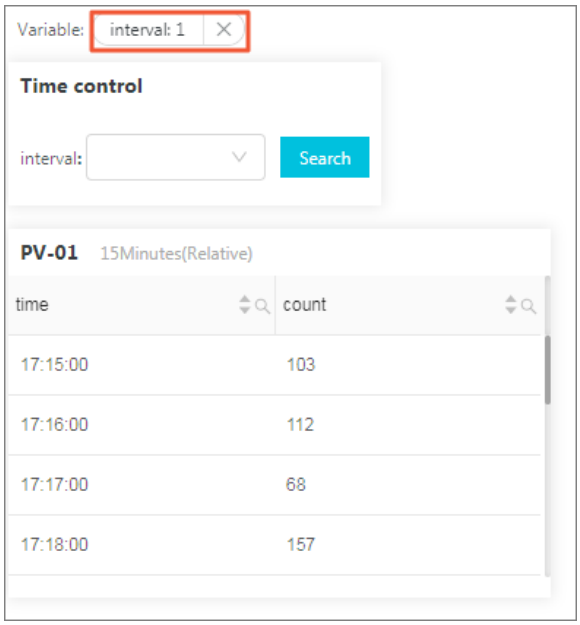
2. Add the chart to a dashboard, and select `60` generate a variable with the name `interval`.



3. Add a filter on the dashboard page.
   - **Type**: Select **Replace Variable**.
   - **Key**: Enter `interval`.
   - **Static List Items**: Add `1` and `120` as values of the key. The default unit is seconds.
4. In the Filter section of the dashboard, select `1` from the Interval drop-down list to view statistics by second.

   The following statement shows the query statement after the placeholder variable is replaced:

```
* | SELECT date_format(__time__ - __time__ % 1, '%H:%i:%s') as time, count(1) as count GROUP BY time ORDER BY time
```



Example 2: Switch between request methods

You can add dynamic values to a filter to dynamically switch between request methods. In example 1, the query statement starts with an asterisk (`*`), which means no condition is set to filter the query results and all logs are queried. You can add another filter to view the PV data of different request `methods`.

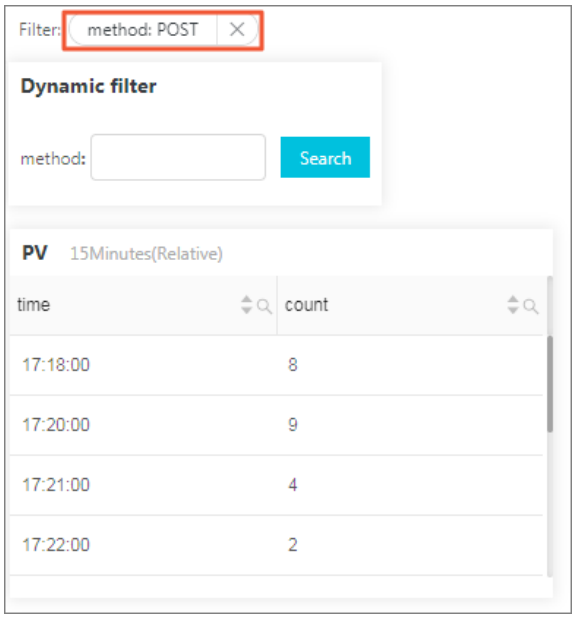
1. Add a filter on the dashboard and turn on the **Add Dynamic List Item** switch.

   Set the parameters as follows:
   - **Type**: Select Filter.
   - **Key**: Enter `method`.
   - **Select Logstore**: Select the Logstore to which the dashboard belongs.
   - **Add Dynamic List Item**: Enter a query statement to obtain the relevant dynamic values, and then click OK.

2. In the Filter section of the dashboard, select `POST` from the method drop-down list.

   Only the PV data whose `method` is `POST` is displayed in the chart. The query statement is changed as follows:

```
(*) and (method: POST) | SELECT date_format(__time__ - __time__ % 60, '%H:%i:%s') as time, count(
1) as count GROUP BY time ORDER BY time
```



28.1.4.11.2.7. Markdown chart

Log Service allows you to add a Markdown chart to a dashboard. In the Markdown chart, you can insert images, links, videos, and other elements to make your dashboard page more intuitive.

Context

You can create different Markdown charts based on your business needs. Markdown charts can make a dashboard more informative. You can insert text such as background information, chart description, notes, extension information, and custom images in a Markdown chart. You can also insert saved searches or dashboard links of other projects to redirect to other query pages.

Scenarios

You can insert links in a Markdown chart to redirect to other dashboard pages of the current project. You can also insert an image corresponding to each link for better recognition. In addition, you can use a Markdown chart to describe parameters of analysis charts.

Procedure

1. Log on to the Log Service console.
2. Click the destination project name.
3. In the left-side navigation pane, click the **Dashboard** icon.
4. In the dashboard list, click the name of the destination dashboard.
5. In the upper-right corner of the dashboard page, click **Edit** to enter the edit mode.
6. In edit mode, drag the Markdown icon from the menu bar to the specified location to create a Markdown chart.
7. Click the created Markdown chart, find the More icon in the upper-right corner of the chart, and click **edit**.

Parameter	Description
Chart Name	The name of the Markdown chart.
Show Border	Specifies whether to show the borders of a Markdown chart. You can turn on the **Show Border** switch to show the borders of a Markdown chart.
Show Title	Specifies whether to show the title of a Markdown chart. You can turn on the **Show Title** switch to show the title of a Markdown chart.
Show Background	Specifies whether to show the background of a Markdown chart. You can turn on the **Show Background** switch to show the background of a Markdown chart.
Query Binding	Specifies whether to bind a query statement to a Markdown chart. You can turn on the **Query Binding** switch and bind a query statement to a Markdown chart. Then, query results are dynamically displayed in the Markdown chart.

8. (Optional)Bind a query statement.
   i. Select the destination Logstore and enter a query statement in the search box. A query statement consists of a search statement and an analytic statement in the format of `search statement|analytic statement`.
   ii. On the Search & Analysis page, click **15 Minutes(Relative)** to set the time range for the query.
   iii. Click **Search** to display the first values of the returned query result.
   iv. Click the plus sign next to a field to insert the corresponding query result to **Markdown Content**.
9. Edit **Markdown Content**.

   Enter Markdown content in the **Markdown content** column. Then, data preview is displayed in real time in the **Show Chart** column on the right. You can modify the Markdown content based on the data preview.

Modify a Markdown chart

- Modify the location and size of a Markdown chart
  i. Click **Edit** in the upper-right corner of the **Dashboard** page.
  ii. Drag the Markdown icon to the specified location on the dashboard and drag the lower-right corner of the chart to adjust its size.
  iii. Click **Save** in the upper-right corner of the dashboard page to save the modification.
- Modify the title of a Markdown chart
  i. Click **Edit** in the upper-right corner of the **Dashboard** page.
  ii. Click the specified Markdown chart, find the More icon in the upper-right corner of the chart, and click **Edit**.
  iii. Enter a new title in the **Chart name** field and then click **OK**.
  iv. Click **Save** in the upper-right corner of the dashboard page. In the dialog box that appears, click **OK**.
- Modify the content of a Markdown chart
  i. Click **Edit** in the upper-right corner of the **Dashboard** page.
  ii. Click the specified Markdown chart, find the More icon in the upper-right corner of chart, and then click **Edit**.
  iii. Modify the chart content, and then click **OK**.
  iv. Click **Save** in the upper-right corner of the dashboard page. In the dialog box that appears, click **OK**.
- Delete a Markdown chart
  i. Click **Edit** in the upper-right corner of the **Dashboard** page.

ii. Click the specified Markdown chart, find the More icon in the upper-right corner of chart, and then click **Delete**.

iii. Click **Save** in the upper-right corner of the dashboard page. In the dialog box that appears, click **OK**.

Common Markdown syntax

- Title

  Markdown syntax:

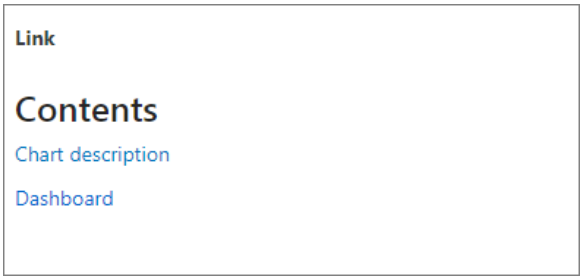
```
# Level 1 heading
## Level 2 heading
### Level 3 heading
```

- Link

  Markdown syntax:

```
### Contents
[Chart description](https:// xxx)
[Dashboard](https:// xxx)
```

  Link preview



- Image

  Markdown syntax:

```
<div align=center>
![ Alt txt][id]
With a reference later in the document defining the URL location
[id]: https://octodex.github.com/images/dojocat.jpg  "The Dojocat"
```

- Special tag

  Markdown syntax:

```
---
__Advertisement :)__
==some mark==   `some code`
> Classic markup: :wink: :crush: :cry: :tear: :laughing: :yum:
>> Shortcuts (emoticons): :-)  8-) ;)
__This is bold text__
*This is italic text*
---
```

28.1.5. Alerts

28.1.5.1. Overview

Log Service enables you to configure alerts for charts on a dashboard to monitor the service status in real time.

You can configure alerts on the **Search & Analysis** page of a Logstore or on a **Dashboard** page. When you configure an alert, you must configure the alert name, trigger condition, notification method, and other parameters. After you Configure alerts, Log Service checks the query results on the dashboard at an interval and sends an alert notification if the check results meet the specified conditions. In this way, Log Service facilitates real-time monitoring of the service status.

Limits

Item	Description
Associated charts	The number of charts that can be associated with an alert ranges from 1 to 3.
String	If the length of a string exceeds 1,024 characters, only the first 1,024 characters are computed during a query.
Conditional expression	• The conditional expression must be 1 to 128 characters in length. • The conditional expression is evaluated based on the first 100 log entries returned for a query. • The conditional expression can be evaluated up to 1,000 times. If the conditional expression is not matched, the alert is not triggered.
Search Period	The time range of each query statement cannot exceed 24 hours.

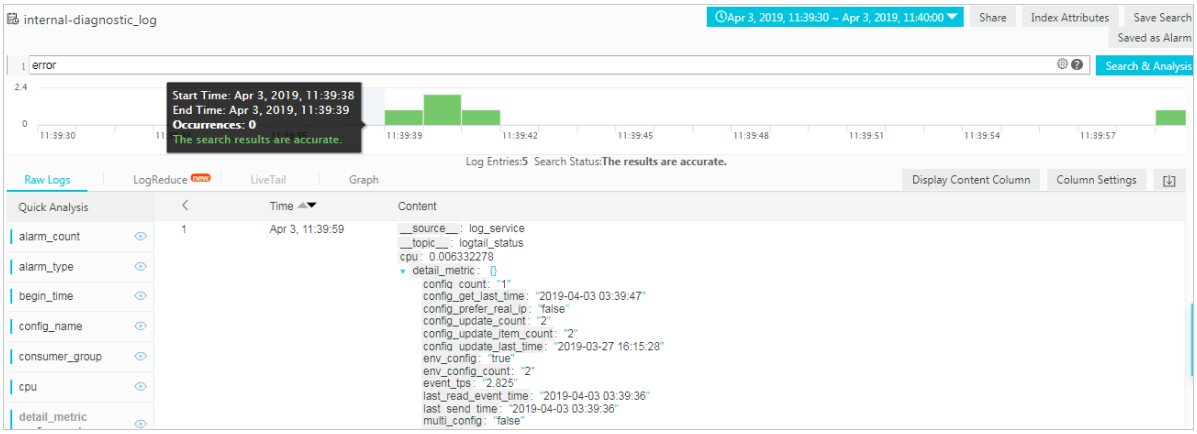
Query statement in an alert

An alert is configured based on analysis charts on a dashboard. An analysis chart is a visualized query result of a query statement. A query statement can include a search statement and an analytic statement.

- If you use only a search statement for a query, log data that matches the search condition is returned.
- If you include search and analytic statements for a query, log data that matches the search condition is analyzed before being returned.
- Search statement

  For example, you want to query the data that contains "error" information in the last 15 minutes. The search statement is error. A total of 154 log entries are retrieved. Each log entry consists of key-value pairs. You can set an alert rule for the value of a key.

  > **Note** If over 100 log entries are returned for a query, only the first 100 log entries are used for evaluating the conditions set in an alert. An alert is triggered when any of the first 100 log entries returned meets the conditions.

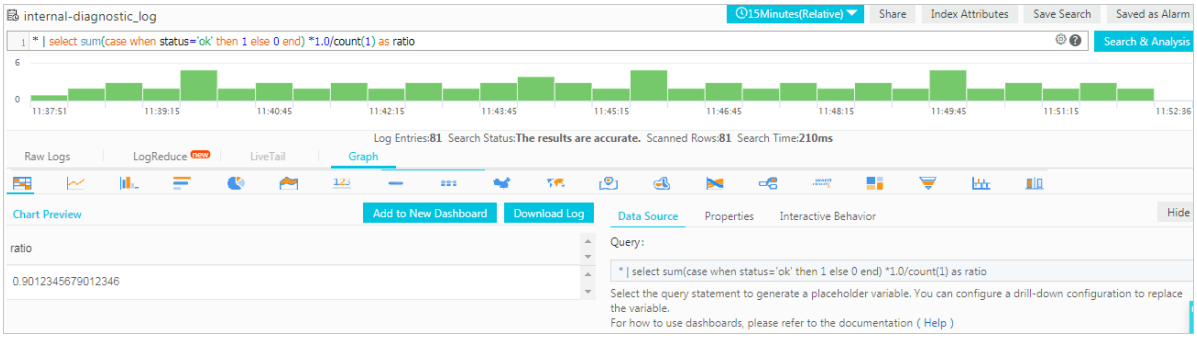


- Search and analytic statement

For example, the following statement queries the ratio of log entries whose status field value is ok to all log entries. For more information about query syntax, see Query syntax.

```
* | select sum(case when status='ok' then 1 else 0 end) *1.0/count(1) as ratio
```

If you set the trigger condition of an alert to `ratio < 0.9`, the alert is triggered when the ratio of log entries whose status field value is ok to all log entries is less than 90%.



28.1.5.2. Configure an alarm

28.1.5.2.1. Configure alerts

Log Service allows you to configure alerts on the Search & Analysis page of a Logstore or on a dashboard page. If the trigger condition of an alert is met, the alert is triggered and a notification is sent to specified recipients.

Prerequisites

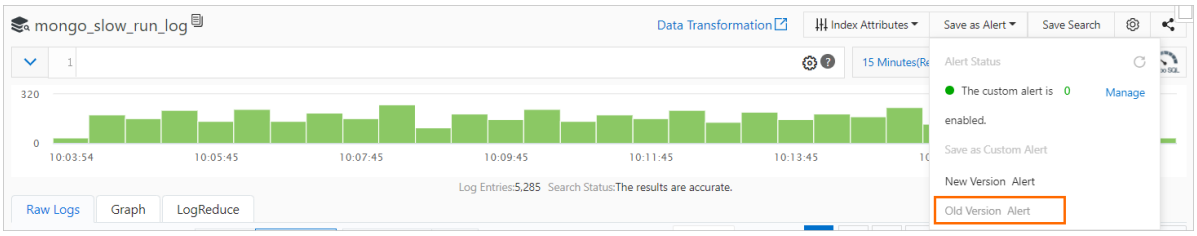
- Log data is collected.
- The indexing feature is enabled and indexes are configured. For more information, see Enable the index feature and configure indexes for a Logstore.

Context

Alerts are configured based on analysis charts. When you view an analysis chart, you can add the chart to a dashboard and configure an alert for the chart. You can also configure an alert for the existing charts on a dashboard.

- Create a chart and configure an alert for the chart

  You can save the results of a query statement as a chart on a dashboard, and configure an alert for the chart. When you configure an alert on the Search & Analysis page, you must specify the name of the dashboard on which the chart is saved and the chart name.



- Configure an alert for existing charts on a dashboard.

  You can configure an alert for one or more charts on a dashboard at a time. When you configure an alert for multiple charts, you can specify a conditional expression for each chart and combine the conditional expressions into the trigger condition for the alert.

This topic describes how to configure an alert for existing charts on a dashboard.

> **Note** If an alert is configured for a chart on a dashboard and you update the search and analytic statement of the chart, you must update the search and analytic statement in the alert configuration. For more information, see Modify an alert.

For information about example alert configurations, see FAQ about alerts.

Procedure

1. Log on to the Log Service console.
2. In the Projects section, click the name of the project in which you want to create a Logstore.
3. In the left-side navigation pane, click the **Dashboard** icon.
4. Click the target dashboard name.
5. In the upper-right corner of the dashboard, choose **Alerts > Create**.
6. In the Create Alert wizard, configure an alert and click **Next**.

   The following table describes the configuration parameters of an alert.

Parameter	Description
Alert Name	The name of the alert. The name must be 1 to 64 characters in length.
Associated Chart	The chart that is associated with the alert. The **Search Period** parameter specifies the time range of log data that Log Service reads when you query data. You can select a relative time or a time frame. For example, if you set **Search Period** to 15 minutes (relative) and start the query at 14:30:06, Log Service reads the log data that was written from 14:15:06 to 14:30:06. If you set **Search Period** to 15 minutes (time frame) and start the query at 14:30:06, Log Service reads the log data that was written from 14:15:00 to 14:30:00. To associate the alert with multiple charts, you must separately add and configure the charts. The number before the chart name indicates the sequence number of the chart in the alert configuration. You can use the sequence number to associate a chart in the trigger condition.
Frequency	The time interval at which Log Service executes the alert.
Trigger Condition	The conditional expression that determines whether to trigger the alert. If the condition is met, Log Service sends an alert notification based on the specified **Check Frequency** and **Notification Interval**. For example, you can enter `pv%100 > 0 && uv > 0` in the Trigger Condition field. **Note** In the conditional expression, you can use `$sequence number` to differentiate charts. For example, you can use `$0` to identify the chart whose sequence number is 0.
Advanced	

Parameter	Description
Notification Trigger Threshold	If the number of times that the trigger condition is met exceeds this threshold and the specified **Notification Interval** elapses, Log Service sends an alert notification to the specified recipients. The default value of **Notification Trigger Threshold** is 1. This value indicates that each time the specified **Trigger Condition** is met, Log Service checks **Notification Interval** to determine whether to send notifications. You can set a custom value. This way, Log Service sends an alert notification to the specified recipient only after the trigger condition is met multiple times. For example, if you set the value to 100, Log Service checks **Notification Interval** only after the trigger condition is met 100 times. If the **Notification Trigger Threshold** and **Notification Interval** are reached, Log Service sends an alert notifications to the specified recipients. The overall count is then reset to zero. If Log Service fails to check log data due to exceptions such as network failures, the overall count does not change.
Notification Interval	The time interval at which Log Service sends an alert notification. If the number of times that the trigger condition is met exceeds the specified **Notification Trigger Threshold** and the specified notification interval elapses, Log Service sends an alert notification to the specified recipients. If you set this parameter to 5 minutes, you can receive up to one alert notification every 5 minutes. The default value is No Interval. **Note** You can use the Notification Trigger Threshold and Notification Interval parameters to control the number of alert notifications that you receive.

7. Set the notification method.

   Notifications can be sent to a custom webhook address in a specified format. To use this notification method, you must set the **Request URL**, **Request Method**, and **Request Content** parameters. For more information, see Notification methods.

   - **Request URL**: a custom webhook address, for example, https://webhook.com/notify.
   - **Request Method**: the request method. Request methods include GET, PUT, POST, DELETE, and OPTIONS.
   - **Request Content**: the content of the notification. The content must be 1 to 500 characters in length. Template variables are supported.
8. Click **OK**.

28.1.5.2.2. Grant permissions on alerts to a RAM user

This topic describes how to grant a RAM user the permissions to enable the alerting feature.

Context

Grant a RAM user the permissions only to create and modify alerts. Create a custom authorization policy, and apply the policy to the RAM user. For more information, see Procedure in this topic.

Procedure

1.
2. Create a RAM role.
3. Create a permission policy.

Use the following policy and replace the `<Project name>` with the actual project name.

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "log:CreateLogStore",
        "log:CreateIndex",
        "log:UpdateIndex"
      ],
      "Resource": "acs:log:*:*:project/<Project name>/logstore/internal-alert-history"
    },
    {
      "Effect": "Allow",
      "Action": [
        "log:CreateDashboard",
        "log:CreateChart",
        "log:UpdateDashboard"
      ],
      "Resource": "acs:log:*:*:project/<Project name>/dashboard/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "log:*"
      ],
      "Resource": "acs:log:*:*:project/<Project name>/job/*"
    }
  ]
}
```

4. Create a user.
5. Create a RAM user group.
6. Add a RAM user to a RAM user group
7. Grant permissions to a RAM role.

28.1.5.2.3. Notification methods

This topic describes the available notification methods that you can configure for an alert in Log Service.

WebHook-Custom

You can set the notification method to WebHook-Custom. When an alert is triggered, Log Service sends an alert notification to a specified webhook URL by using a specified method.

> **Note** The timeout period of the **WebHook-Custom** notification method is five seconds. If no response is received within five seconds after a notification request is sent, the notification request is deemed to be failed.

1. Configure an alert in the Log Service console. For more information, see Configure alerts. Select **WebHook-Custom** from the Notifications drop-down list.
2. Enter your custom webhook URL in the **Request URL** field. Select a **Request Method**.
3. (Optional)Click **Add Request Headers** to add request header fields.

   By default, the request header contains the field `Content-Type: application/json;charset=utf-8`. You

can add request header fields based on your business needs.

4. Enter the notification content in the **Request Content** field.

   When an alert is triggered, Log Service sends the specified notification content to the custom webhook URL by using the specified method.

5. Click **Submit**.

Template variables

You must set **Content** for each notification method. In the notification content, you can reference some template variables in the ${fieldName} format for the alert. When sending an alert notification, Log Service replaces the template variables referenced in the **Content** field with real values. For example, it replaces ${Project} with the name of the project to which the alert belongs.

> **Note** You must reference valid variables. If a referenced variable does not exist or is invalid, Log Service processes the variable as a null string. If the value of a referenced variable is of the object type, the value is converted and displayed as a JSON string.

The following table lists all the supported variables and reference methods.

Variable	Description	Example	Reference example
Aliuid	The ID of the Apsara Stack tenant account to which the project belongs.	1234567890	The alert configured by the user ${Aliuid} is triggered.
Project	The project to which the alert belongs.	my-project	The alert configured in the project ${Project} is triggered.
AlertID	The unique ID of the alert.	0fdd88063a611aa114938f9371daeeb6-1671a52eb23	The ID of the alert is ${AlertID}.
AlertName	The name of the alert, which must be unique in a project.	alert-1542111415-153472	The alert ${AlertName} is triggered.
AlertDisplayName	The display name of the alert.	My alert	The alert ${AlertDisplayName} is triggered.
Condition	The conditional expression that triggers the alert. Each variable in the conditional expression is replaced with the value that triggers the alert. The value is enclosed in brackets [].	[5] > 1	The conditional expression that triggers the alert is ${Condition}.
RawCondition	The original conditional expression that triggers the alert. Variables in the conditional expression are not replaced.	count > 1	The original conditional expression that triggers the alert is ${RawCondition}.
Dashboard	The name of the dashboard with which the alert is associated.	mydashboard	The alert is associated with the dashboard ${Dashboard}.
DashboardUrl	The URL of the dashboard with which the alert is associated.	https://sls.console.aliyun.com/next/project/myproject/dashboard/mydashboard	The URL of the dashboard associated with the alert is ${DashboardUrl}.

Variable	Description	Example	Reference example
FireTime	The time when the alert is triggered.	2018-01-02 15:04:05	The alert is triggered at ${FireTime}.
FullResultUrl	The URL used to query the history records that an alert rule was executed.	https://sls.console.aliyun.com/next/project/my-project/logsearch/internal-alert-history?endTime=1544083998&queryString=AlertID%3A9155ea1ec10167985519fccede4d5fc7-1678293caad&queryTimeType=99&startTime=1544083968	Click ${FullResultUrl} to view details.

Variable	Description	Example	Reference example

Results	The parameters and results of each log data query. The value is of the array type. For information about parameters in the Results field, see Fields in alert log entries.
	
	Note A maximum of 100 alert notifications can be sent.
		<pre>[
		  {
		    "EndTime": 1542507580,
		    "FireResult": {
		      "__time__": "1542453580",
		      "count": "0"
		    },
		    "LogStore": "test-logstore",
		    "Query": "* | SELECT COUNT(*) as count",
		    "RawResultCount": 1,
		    "RawResults": [
		      {
		        "__time__": "1542453580",
		        "count": "0"
		      }
		    ],
		    "StartTime": 1542453580
		  }
		]</pre>
			The first query starts at ${Results[0].StartTime} and ends at ${Results[0].EndTime}. The alert has been triggered ${Results[0].FireResult.count} times.
			
			Note In this example, the number 0 indicates the sequence number of the chart or the search and analytic statement. For more information, see How can I check the sequence number of a chart?

28.1.5.3. Modify and view an alarm

28.1.5.3.1. Modify an alert

This topic describes how to modify an alert. After you create an alert, you can modify the alert and then update the alert. To modify an alert associated with a search statement, you can directly modify the search statement.

Precautions

- You can modify only search statements with which alerts are associated. You cannot modify search statements to search and analytic statements, which are in the format of `search statement|analytic statement`.

  For example, after you associate the `request_method: GET` statement with an alert, you can modify the statement to `error`, but you cannot modify the statement to `error| select count(1) as c`.
- To modify an alert, you can click **Modify Settings** on the **Alert Overview** page, or choose **Alerts > Modify** on the associated dashboard.

Modify the search statement associated with an alert

If you associate a search statement with an alert, you can modify the search statement to modify the alert.

1. Log on to the Log Service console.
2. Click a project name.
3. In the left-side navigation pane, click the **Dashboard** icon.
4. In the dashboard list, click the name of the target dashboard.
5. On the dashboard, choose **Alerts > Modify**.
6. Find the search statement, and then click  .

   You can modify only search statements with which alerts are associated. You cannot modify search statements to search and analytic statements, which are in the format of `search statement|analytic statement`.
7. On the dialog box that appears, enter a new search statement, click **Preview**, and then click **OK** after the search statement is verified.
8. Modify other parameters specific to your environment, such as **Frequency** and **Trigger Condition**, and then click **Next**.
9. Set the notification method, and then click **Submit**.

Modify the chart associated with an alert

After you create an alert, you can modify the chart associated with the alert to modify the alert.

1. In the dashboard list, click the name of the target dashboard.
2. On the dashboard, choose **Alerts > Modify**.
3. Find the **Associated Chart**, and then click  next to **Query**.
4. On the dialog box that appears, enter a new query statement, click **Preview**, and then click **OK** after the query statement is verified.
5. Modify other parameters specific to your environment, such as **Frequency** and **Trigger Condition**, and then click **Next**.
6. Set the notification method.
7. Click **Submit**. The new settings take effect immediately.

28.1.5.3.2. View history alerts

This topic describes how to view history alerts in the Log Service console. Log Service records alerts as log data and creates a dashboard to display alert details.

View history alerts in the Logstore

When you create an alert, Log Service creates a Logstore named **internal-alert-history** for the project to which the alert belongs. A log entry is generated and written to the Logstore each time the alert rule is executed, regardless of whether the alert is triggered. For more information about the fields in the log entry, see Fields in alert log entries.

> **Note** The Logstore does not incur fees and cannot be deleted or modified. Each alert log entry is retained in the Logstore for seven days.

1. Log on to the Log Service console.
2. Click a project name.
3. Click the icon next to the **internal-alert-history** Logstore, and then select **Search & Analysis**.
4. On the page that appears, query alert log entries based on your needs.

View history alerts on the dashboard

After you create an alert, Log Service creates a dashboard named **internal-alert-analysis** for the project to which the alert belongs. The dashboard displays the statistics of all previous alerts, including the number of triggered alerts, percentage of successful alerts and notifications, and top 10 alerts whose alert rules are executed.

> **Note** The dashboard cannot be deleted or modified.

1. In the left-side navigation pane, click the **Dashboard** icon.
2. Click **Alert History Statistics** to open the dashboard page.

   On the **Alert History Statistics** dashboard, the details of history alerts are displayed, including whether the alerts are triggered, why the alerts are triggered, error information, and other information.

28.1.5.3.3. Manage an alert

This topic describes how to manage an alert after you create the alert.

Context

You can disable, enable, modify, and delete the alert, or view the details of the alert such as the update time.

View the details of an alert

1. Log on to the Log Service console.
2. Click a project name.
3. In the left-side navigation pane, click the **Alerts** icon.
4. In the alert list, click the name of the target alert.

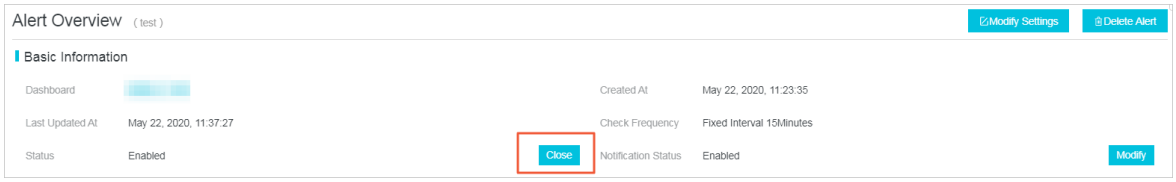
   On the **Alert Overview** page, you can view the details of the alert, such as the dashboard, creation time, last update time, check frequency, alert status, and notification status.

Disable and enable an alert

After you create an alert, you can disable or enable the alert. If you disable an alert, Log Service no longer checks the alert or send alert notifications.

1. In the left-side navigation pane, click the **Alerts** icon.
2. In the alert list, click the name of the target alert.

   On the **Alert Overview** page, click **Enable** or **Close** in the **Alert Status** field.

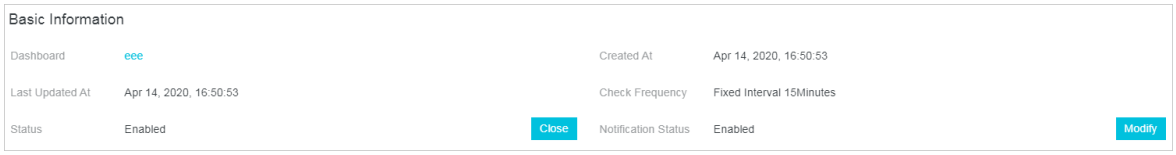


Disable and enable alert notifications

You can disable notifications of enabled alerts. After you disable notifications of an alert, alert notifications are not sent even if the trigger condition is met.

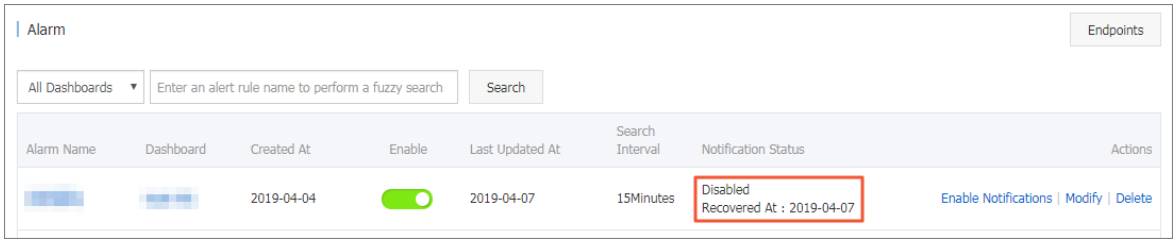
1. In the left-side navigation pane, click the **Alerts** icon.
2. In the alert list, click the name of the target alert.

   On the **Alert Overview** page, click **Modify** in the **Notification Status** field.



3. Set the time range for which notifications are disabled, and then click **Confirm**.

   After you disable alert notifications, you can view the time when alert notifications resume in the **Notification Status** field. You can click **Modify** in the **Notification Status** field to manually resume alert notifications.



Delete an alert

You cannot recover a deleted alert. Proceed with caution when you delete an alert.

1. In the left-side navigation pane, click the **Alerts** icon.
2. In the alert list, click the name of the target alert.
3. On the **Alert Overview** page, click **Delete Alert**.
4. In the dialog box that appears, click **OK**.

28.1.5.4. Relevant syntax and fields for reference

28.1.5.4.1. Conditional expression syntax of an alert

This topic describes how to configure a conditional expression for an alert in Log Service. An alert is triggered if the conditional expression configured for the alert is met.

In determining whether the conditional expression of an alert is met, the results of query statements configured for the alert are used as the inputs and the log fields in the conditional expression are used as the variables. If the conditional expression is met, the alert is triggered.

Limits

- Negative numbers must be enclosed with parentheses (), for example, x +(-100)<100.
- Numeric data is treated as 64-bit floating-point numbers. If a comparison is performed, errors may occur.
- A variable can contain only letters and digits and must start with a letter.
- A conditional expression can be up to 128 characters in length.
- A conditional expression can be evaluated up to 1,000 times. If an alert is configured for multiple charts and the conditional expression of the alert is not met after 1,000 times of evaluation, the alert is not triggered.
- A maximum of three charts can be associated with an alert.
- An alert is triggered if and only if the Boolean value of its conditional expression is true. For example, the result of the expression 100+100 is 200, which cannot trigger the alert.
- `true`, `false`, `$`, and `.` are reserved and cannot be used as variables.

Basic syntax

The following table lists the syntax supported in a conditional expression.

Syntax	Description	Examples
Basic operators	Supports the addition operator (+), subtraction operator (-), multiplication operator (*), division operator (/), and modulus operator (%).	x*100+y>200 x%10>5
Comparison operators	Supports eight comparison operators, including the greater than operator (>), greater than or equal to operator (>=), less than operator (<), less than or equal to operator (<=), equal to operator (==), not equal to operator (! =), match operator (=~), and mismatch operator (! ~). **Note** • Backslashes (/) must be escaped. • Regular expressions support the RE2 syntax.	x >= 0 x < 100 x <= 100 x == 100 x == "foo" Regular expression: x =~ "\\w +"
Logical operators	Supports the logical operators AND (&&) and OR (||).	x >=0&&y <=100 x > 0 || y > 0
Logical negation	Supports the logical negation operator (!).	!(a < 1 && a > 100)

Syntax	Description	Examples
Numeric constants	Numeric constants are processed as 64-bit floating-point numbers.	x > 100
String constants	String constants are formatted in a string enclosed in single quotation marks ('').	foo == 'string'
Boolean constants	Supports true and false.	(x > 100) == true
Parentheses	Parentheses () can be used to enforce precedence order.	x*(y+100)>100
contains function	Determines whether a substring is included in a string. For example, if the result of the contains(field, 'xxxx') function is true, the field string includes the xxxx substring.	contains(foo, 'hello')

Conditional expression for results of multiple query statements

- Syntax

  If you configure an alert for multiple charts, the variables in a conditional expression must be prefixed. In this way, you can specify from which query result to obtain the corresponding values of the variables when it evaluates your expression. The format is `$N.fieldname`, where N is the sequence number of the query statement. You can configure up to three query statements in an alert. Therefore the value range of N is 0 to 2. For example, $0.foo indicates to obtain the value of the foo field returned from the first query statement. If you configure one query statement in an alert, you do not need to specify the prefix.

  > **Note** How can I view the sequence number of a query statement?
  >
  > In the Alert Configuration step, the **Associated Chart** parameter specifies the sequence number of each query statement (chart). The first query statement is numbered 0, the second query statement is numbered 1, and the third statement is numbered 2.

- Evaluate a conditional expression

  If multiple query results are returned, the variables specified in the conditional expression determine how to use the results to evaluate the conditional expression. For example, if you configure three query statements in an alert, x, y, and z log entries are returned when you execute each of the three statements. The conditional expression that you configure for the alert is $0.foo > 100 && $1.bar < 100. Then the first two query results are used to evaluate the conditional expression. A maximum of **x × y** times of evaluation (or 1,000 if x × y is greater than 1,000) is performed. If the conditional expression is met within the maximum times of evaluation, true is returned. Otherwise, false is returned.

Operations

> **Note**
>
> - 64-bit floating-point numbers are used in a conditional expression.
> - Each string constant must be enclosed in single quotation marks ('') or double quotation marks (""), for example, 'string', and "string".
> - Boolean values include true and false.

Operator	Operation		
	Operation between variables	Operation between non-string constants and variables	Operation between string constants and variables
Basic operators, including the addition operator (+), subtraction operator (-), multiplication operator (*), division operator (/), and modulus operator (%)	The left and right operands are converted to numbers before being operated.		Unsupported.
Comparison operators, including the greater than operator (>), greater than or equal to operator (>=), less than operator (<), less than or equal to operator (<=), equal to operator (==), not equal to operator (! =)	Operations are performed based on the following priorities: 1. The left and right operands are converted to numbers before being operated based on the numerical order. If the conversion fails, 2. operands are operated based on the alphabetical order of strings.	The left and right operands are converted to numbers before being operated based on the numerical order.	The left and right operands are operated based on the alphabetical order of strings.
Regular expression match operator (=~) and regular expression mismatch operator (! ~)	The left and right operands are operated based on the alphabetical order of strings.	Unsupported.	The left and right operands are operated based on the alphabetical order of strings.
Logical operators, including AND (&&) and OR (||)	These two operators cannot be directly used on the fields in query results. The left and right operands must be sub-expressions, and the values of the sub-expressions are of the Boolean type.		
Logical negation (!)	This operator cannot be directly used on the fields in query results. The left and right operands must be sub-expressions, and the values of the sub-expressions are of the Boolean type.		

Operator	Operation		
	Operation between variables	Operation between non-string constants and variables	Operation between string constants and variables
contains function	The left and right operands are operated based on the alphabetical order of strings.	Unsupported.	The left and right operands are operated based on the alphabetical order of strings.
Parentheses ()	Parentheses () enforce precedence order.		

28.1.5.4.2. Fields in alert log entries

After you configure an alert, Log Service automatically creates a Logstore to store log entries that are generated when alert rules are executed and notifications are sent. This topic describes fields in alert log entries.

Fields

Field	Description	Example
AlertDisplayName	The display name of an alert.	Test alert rules
AlertID	The unique ID of an alert.	0fdd88063a611aa114938f9371daeeb6-1671a52eb23
AlertName	The unique name of an alert in a project.	alert-1542111415-153472
Condition	The conditional expression configured for an alert.	$0.count > 1
Dashboard	The dashboard associated with an alert.	my-dashboard
FireCount	The cumulative times that an alert has been triggered since the last notification was sent.	1
Fired	Indicates whether an alert was triggered. Valid values: true and false.	true
LastNotifiedAt	The time when the last notification was sent. The time is displayed in a Unix timestamp.	1542164541
NotifyStatus	The status of a notification. Valid values: • Success: indicates that a notification was successfully sent. • Failed: indicates that a notification failed to be sent. • NotNotified: indicates that no notification was sent. • PartialSuccess: indicates that the notification sending partially succeeded.	Success

Field	Description	Example
Reason	The reason that a notification failed to be sent or no notification was sent.	result type is not bool

Results	The parameters and results of each log data query. The value is of the array type. For information about parameters in the Results field, see Parameters in the Result field.	<pre>[
		  {
		  "EndTime": 1542334900,
		  "FireResult": null,
		  "LogStore": "test-logstore",
		  "Query": "* | select count(1) as count",
		  "RawResultCount": 1,
		  "RawResults": [
		  {
		  "__time__": "1542334840",
		  "count": "0"
		  }
		  ],
		  "StartTime": 1542334840
		  }
		]</pre>

Status	The execution result of an alert. Valid values: Success and Failed.	Success

Parameters in the Result field

Parameter	Description	Example
Query	The query statement that is configured in an alert.	* | select count(1) as count
LogStore	The target Logstore of a query.	my-logstore
StartTime	The time when a query starts.	2019-01-02 15:04:05
StartTimeTs	The time when a query starts. The time is in the Unix timestamp format.	1542334840
EndTime	The time when a query ends.	2019-01-02 15:19:05
EndTimeTs	The time when a query ends. The time is in the Unix timestamp format. The actual query time range is `[StartTime, EndTime)`.	1542334900

Parameter	Description	Example

RawResults	The raw query result. The parameter value is formatted in an array where each element is a log entry. The maximum length of the array is 100.	<pre>[
		  {
		  "__time__": "1542334840",
		  "count": "0"
		  }
		]</pre>

RawResultsAsKv	The query result that is formatted in key-value pairs. **Note** This parameter can only be used as a template variable. It is not saved to a Logstore.	[foo:0]
RawResultCount	The number of log entries that are returned in the RawResults parameter.	1

FireResult	The log entry that records the triggering of an alert. If an alert is not triggered, the parameter value is null.	<pre>{
		  "__time__": "1542334840",
		  "count": "0"
		}</pre>

FireResultAsKv	The log entry that records the triggering of an alert, formatted in key-value pairs. **Note** This parameter can only be used as a template variable. It is not saved to a Logstore.	[foo:0]

28.1.6. Real-time consumption

28.1.6.1. Overview

Log Service allows you to consume log data by using multiple methods.

After data is collected to LogHub, you can consume the log data by using two methods. The following table describes the methods.

Method	Scenarios	Timeliness	Retention period
Real-time consumption (LogHub)	Real-time computing	Real time	Custom

Method	Scenarios	Timeliness	Retention period
Indexing and query (LogSearch)	Online query of hot data	Real time	Custom

Real-time consumption

LogHub allows you to pull log data and consume the data in real time. The following procedure describes how log data in a shard is consumed:

1. Obtain a cursor based on the start time and end time of data consumption.
2. Read log data based on the cursor and step parameters and return the position of the next cursor.
3. Move the cursor to continuously consume log data.

- Consume log data by using an SDK

  You can use Log Service SDKs in multiple programming languages such as Java, Python, and Go to consume log data.

- Consume log data by using consumer groups

  Log Service provides an advanced method that allows you to consume logs by using consumer groups. A consumer group is a lightweight computing framework that allows multiple consumers to consume data from a Logstore at the same time. The consumers in a consumer group are automatically allocated shards. Data is consumed in order based on the time when it is written to the Logstore. In addition, the consumers can use checkpoints to resume consumption from a breakpoint. You can use consumer group SDKs in multiple programming languages such as Go, Python, and Java to consume log data.

- Log consumption by using real-time stream processing systems
  - Use Spark Streaming clients to consume log data.
  - Use Storm spouts to consume log data.
  - Use Flink Connector to consume log data.

- Log consumption by using open-source services

  Use Flume to consume log data and import log data to Hadoop file system (HDFS) instances.

Log search and analytics

- You can query log data in the Log Service console.
- You can also query log data by using an SDK or the API of Log Service. Log Service provides an HTTP-based RESTful API. You can call all log query API operations that are provided by Log Service.

28.1.6.2. Consume log data

Log Service provides SDKs in various programming languages, such as Java, Python, and Go. You can use an SDK to consume log data.

Use an SDK to consume log data

The following example shows how to use the SDK for Java to consume log data in a shard:

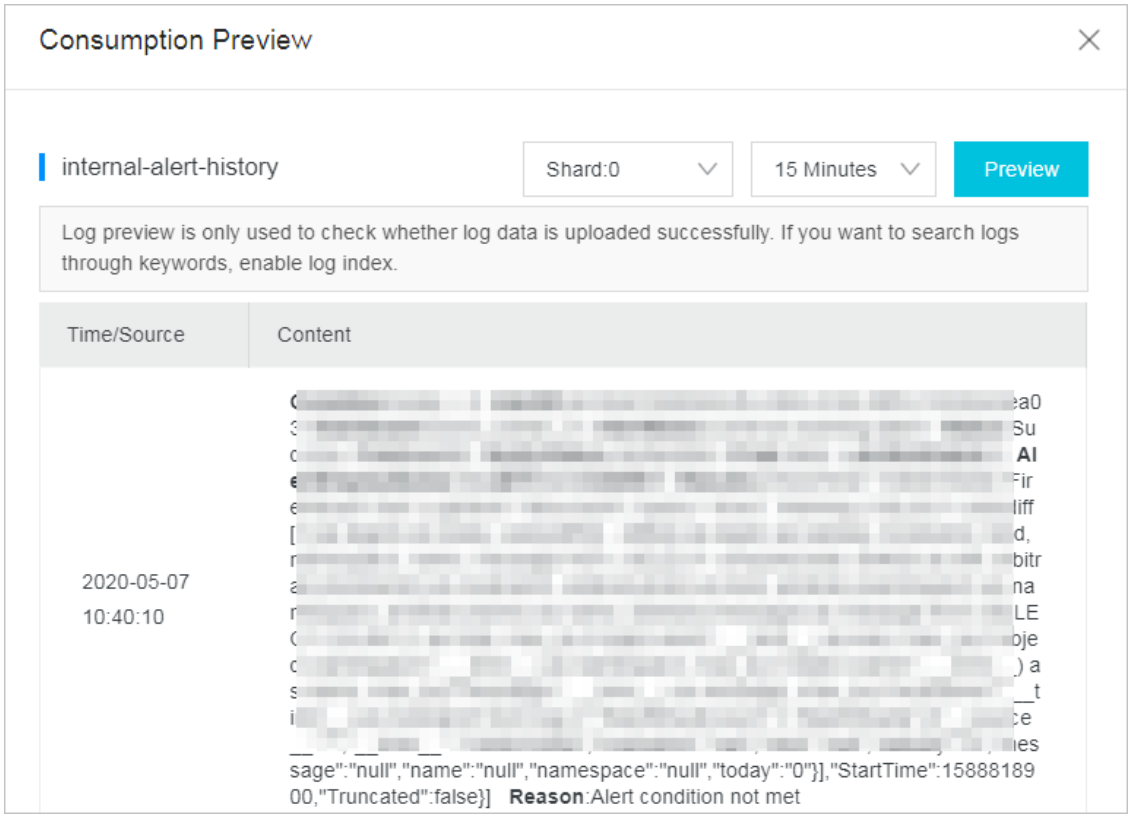
```
Client client = new Client(host, accessId, accessKey);
    String cursor = client.GetCursor(project, logStore, shardId, CursorMode.END).GetCursor();
    System.out.println("cursor = " +cursor);
    try {
      while (true) {
        PullLogsRequest request = new PullLogsRequest(project, logStore, shardId, 1000, cursor);
        PullLogsResponse response = client.pullLogs(request);
        System.out.println(response.getCount());
        System.out.println("cursor = " + cursor + " next_cursor = " + response.getNextCursor());
        if (cursor.equals(response.getNextCursor())) {
            break;
                }
        cursor = response.getNextCursor();
        Thread.sleep(200);
      }
    }
    catch(LogException e) {
      System.out.println(e.GetRequestId() + e.GetErrorMessage());
    }
```

Preview log data in the Log Service console

Log preview is a way of log data consumption. To preview log data that is stored in a Logstore in the Log Service console, perform the following steps:

1. Log on to the Log Service console.
2. In the Projects section, click the target project.
3. In the Logstore list, find the target Logstore, click the icon next to the Logstore, and then select **Consumption Preview**.
4. In the Consumption Preview dialog box, select a shard, set a time range, and then click **Preview**.

   The log preview page displays the log data of the first 10 packets in the specified time range.



28.1.6.3. Consumption by consumer groups

28.1.6.3.1. Use consumer groups to consume log data

Log consumption by using consumer groups

Consumer groups allow you to focus on the business logic during log data consumption. You do not need to consider factors such as Log Service implementation, load balancing among consumers, and failovers that may be introduced when you use an SDK to consume log data.

Terms

The following table describes the terms of consumer groups and consumers.

Term	Description
consumer group	A consumer group consists of multiple consumers. Each consumer in a consumer group consumes different data in a Logstore.
consumer	The consumers in a consumer group consume data from specified data sources. Each consumer name in a consumer group must be unique.

A Logstore has multiple shards. A consumer library allocates shards to consumers in a consumer group based on the following principles:

- Each shard can be allocated to one consumer.
- Each consumer can consume data from multiple shards.

After a new consumer joins a consumer group, the shards allocated to the consumer group are reallocated to each consumer for load balancing. The reallocation is based on the preceding principles and cannot be viewed by users.

A consumer library stores checkpoints. This allows consumers to resume consumption from a breakpoint and avoid repetitive consumption after a program fault is resolved.

Procedure

Log consumption by using consumer groups is implemented in Java or Python. The following procedure takes Java as an example to describe how consumer groups consume log data.

1. Add Maven dependencies.

```
<dependency>
  <groupId>com.google.protobuf</groupId>
  <artifactId>protobuf-java</artifactId>
  <version>2.5.0</version>
</dependency>
<dependency>
  <groupId>com.aliyun.openservices</groupId>
  <artifactId>loghub-client-lib</artifactId>
  <version>0.6.16</version>
</dependency>
```

2. Create a file named main.java.

```
import com.aliyun.openservices.loghub.client.ClientWorker;
import com.aliyun.openservices.loghub.client.config.LogHubConfig;
import com.aliyun.openservices.loghub.client.exceptions.LogHubClientWorkerException;
public class Main {
    // The endpoint of Log Service.
    private static String sEndpoint = "cn-hangzhou.log.aliyuncs.com";
    // The name of a Log Service project.
    private static String sProject = "ali-cn-hangzhou-sls-admin";
    // The name of a Logstore.
    private static String sLogstore = "sls_operation_log";
    // The name of a consumer group.
    private static String sConsumerGroup = "consumerGroupX";
    // The AccessKey pair of an Apsara Stack tenant account or RAM user that is used to consume d
ata.
    private static String sAccessKeyId = "";
    private static String sAccessKey = "";
    public static void main(String[] args) throws LogHubClientWorkerException, InterruptedExcepti
on {
        // The second parameter is the consumer name. Each consumer name in a consumer group must
be unique. However, the names of consumer groups can be the same. When different consumers start
multiple processes on multiple servers to consume the data of a Logstore, you can use a server IP
address to identify a consumer. The ninth parameter maxFetchLogGroupSize indicates the maximum nu
mber of log groups that are retrieved from the server at a time. Valid values: 1 to 1000. You can
use the default value or specify a value based on your requirements.
        LogHubConfig config = new LogHubConfig(sConsumerGroup, "consumer_1", sEndpoint, sProject,
sLogstore, sAccessKeyId, sAccessKey, LogHubConfig.ConsumePosition.BEGIN_CURSOR);
        ClientWorker worker = new ClientWorker(new SampleLogHubProcessorFactory(), config);
        Thread thread = new Thread(worker);
        // The ClientWorker instance automatically runs after the thread is executed and extends
the Runnable interface.
        thread.start();
        Thread.sleep(60 * 60 * 1000);
        // The shutdown function of the ClientWorker instance is called to exit the consumption i
nstance. The associated thread is  automatically stopped.
        worker.shutdown();
        // Multiple asynchronous tasks are generated when the ClientWorker instance is running. W
e recommend that you wait for 30 seconds to ensure that all running tasks exit after shutdown.
        Thread.sleep(30 * 1000);
    }
}
```

3. Create a file named SampleLogHubProcessor.java.

```
import com.aliyun.openservices.log.common.FastLog;
import com.aliyun.openservices.log.common.FastLogContent;
import com.aliyun.openservices.log.common.FastLogGroup;
import com.aliyun.openservices.log.common.FastLogTag;
import com.aliyun.openservices.log.common.LogGroupData;
import com.aliyun.openservices.loghub.client.ILogHubCheckPointTracker;
import com.aliyun.openservices.loghub.client.exceptions.LogHubCheckPointException;
import com.aliyun.openservices.loghub.client.interfaces.ILogHubProcessor;
import com.aliyun.openservices.loghub.client.interfaces.ILogHubProcessorFactory;
import java.util.List;
public class SampleLogHubProcessor implements ILogHubProcessor {
    private int shardId;
    // Record the last persistent checkpoint time.
    private long mLastCheckTime = 0;
    public void initialize(int shardId) {
        this.shardId = shardId;
    }
```

```
    }
    // The main logic of data consumption. All exceptions must be captured and cannot be thrown.
    public String process(List<LogGroupData> logGroups,
                          ILogHubCheckPointTracker checkPointTracker) {
        // Display the retrieved data.
        for (LogGroupData logGroup : logGroups) {
            FastLogGroup flg = logGroup.GetFastLogGroup();
            System.out.println(String.format("\tcategory\t:\t%s\n\tsource\t:\t%s\n\ttopic\t:\t%s\
n\tmachineUUID\t:\t%s",
                    flg.getCategory(), flg.getSource(), flg.getTopic(), flg.getMachineUUID()));
            System.out.println("Tags");
            for (int tagIdx = 0; tagIdx < flg.getLogTagsCount(); ++tagIdx) {
                FastLogTag logtag = flg.getLogTags(tagIdx);
                System.out.println(String.format("\t%s\t:\t%s", logtag.getKey(), logtag.getValue(
)));
            }
            for (int lIdx = 0; lIdx < flg.getLogsCount(); ++lIdx) {
                FastLog log = flg.getLogs(lIdx);
                System.out.println("--------\nLog: " + lIdx + ", time: " + log.getTime() + ", Get
ContentCount: " + log.getContentsCount());
                for (int cIdx = 0; cIdx < log.getContentsCount(); ++cIdx) {
                    FastLogContent content = log.getContents(cIdx);
                    System.out.println(content.getKey() + "\t:\t" + content.getValue());
                }
            }
        }
        long curTime = System.currentTimeMillis();
        // Write checkpoints to the server every 30 seconds. If a ClientWorker instance does not
respond within 30 seconds,
        // a new ClientWorker instance consumes data starting from the last checkpoint. A small a
mount of duplicate data may exist.
        if (curTime - mLastCheckTime > 30 * 1000) {
            try {
                // If the parameter is set to true, checkpoints are synchronized to the server im
mediately. If the parameter is set to false, checkpoints are  locally cached. The default synchro
nization interval of checkpoints is 60 seconds.
                checkPointTracker.saveCheckPoint(true);
            } catch (LogHubCheckPointException e) {
                e.printStackTrace();
            }
            mLastCheckTime = curTime;
        }
        return null;
    }
    // The ClientWorker instance calls this function upon exit. You can perform a cleanup.
    public void shutdown(ILogHubCheckPointTracker checkPointTracker) {
        // Save consumption breakpoints to the server.
        try {
            checkPointTracker.saveCheckPoint(true);
        } catch (LogHubCheckPointException e) {
            e.printStackTrace();
        }
    }
}
class SampleLogHubProcessorFactory implements ILogHubProcessorFactory {
    public ILogHubProcessor generatorProcessor() {
        // Generate a consumption instance.
        return new SampleLogHubProcessor();
    }
}
```

> **Note** Run the preceding code to display all data in a Logstore. If you want multiple consumers to consume data from the same Logstore, you can modify the code based on the comments. You can use the same consumer group name and different consumer names to start new consumption processes.

Limits and troubleshooting

You can create a maximum of 10 consumer groups for each Logstore. The `ConsumerGroupQuotaExceed` error is reported if the number of consumer groups exceeds 10.

We recommend that you configure Log4j for the consumer program to throw error messages within consumer groups. This improves the troubleshooting efficiency. If you save the log4j.properties file to the resources directory and run the program, the following error message appears:

```
[WARN ] 2018-03-14 12:01:52,747 method:com.aliyun.openservices.loghub.client.LogHubConsumer.sampleLog
Error(LogHubConsumer.java:159)
com.aliyun.openservices.log.exception.LogException: Invalid loggroup count, (0,1000]
```

The following example is a typical log4j. properties configuration file:

```
log4j.rootLogger = info,stdout
log4j.appender.stdout = org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target = System.out
log4j.appender.stdout.layout = org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern = [%-5p] %d{yyyy-MM-dd HH:mm:ss,SSS} method:%l%n%m%n
```

Advanced operations

The preceding code is suitable for log consumption in common scenarios. This section describes how to perform advanced operations in other scenarios.

- Consume data that is logged from a certain time point

  LoghubConfig in the preceding code has two constructors:

  ```
  // The value of the consumerStartTimeInSeconds parameter is a UNIX timestamp representing the numb
  er of seconds that have elapsed since 00:00:00 on January 1, 1970, 00:00:00 UTC.
  public LogHubConfig(String consumerGroupName,
                      String consumerName,
                      String loghubEndPoint,
                      String project, String logStore,
                      String accessId, String accessKey,
                      int consumerStartTimeInSeconds);
  // The position parameter is an enumeration variable. LogHubConfig.ConsumePosition.BEGIN_CURSOR in
  dicates that the consumption starts from the earliest data. LogHubConfig.ConsumePosition.END_CURSO
  R indicates that the consumption starts from the latest data.
  public LogHubConfig(String consumerGroupName,
                      String consumerName,
                      String loghubEndPoint,
                      String project, String logStore,
                      String accessId, String accessKey,
                      ConsumePosition position);
  ```

  You can use different constructors based on your requirements. However, if a checkpoint is stored on the server, you must start data consumption from this checkpoint.

- Reset a checkpoint

In scenarios such as data padding or repeated computing, you may need to set the consumption position to a time point for a consumer group. Then data consumption is started from the consumption position. To set the consumption position, you can use either one of the following two methods:

- Delete the consumer group.
  a. Stop the consumption processes.
  b. Delete the consumer group from the Log Service console.
  c. Modify the code to specify the start time point for data consumption.
  d. Restart the consumption processes.
- Use an SDK to reset the start time point of data consumption for the consumer group.
  a. Stop the consumption processes.
  b. Use an SDK to modify the checkpoint.
  c. Restart the consumption processes.

```
public static void updateCheckpoint() throws Exception {
        Client client = new Client(host, accessId, accessKey);
        long timestamp = Timestamp.valueOf("2017-11-15 00:00:00").getTime() / 1000;
        ListShardResponse response = client.ListShard(new ListShardRequest(project, logStore));
        for (Shard shard : response.GetShards()) {
            int shardId = shard.GetShardId();
            String cursor = client.GetCursor(project, logStore, shardId, timestamp).GetCursor();
            client.UpdateCheckPoint(project, logStore, consumerGroup, shardId, cursor);
        }
    }
```

Use a RAM user to access consumer groups

Before you use a RAM user to access consumer groups, you must grant relevant permissions to the RAM user. For more information, see Grant permissions to a RAM role.

The following table lists the actions you can perform as a RAM user.

Action	Resource
log:GetCursorOrData	acs:log:*${regionName}*:*${projectOwnerAliUid}*:project/*${projectName}*/logstore/*${logstoreName}*
log:CreateConsumerGroup	acs:log:*${regionName}*:*${projectOwnerAliUid}*:project/*${projectName}*/logstore/*${logstoreName}*/consumergroup/*
log:ListConsumerGroup	acs:log:*${regionName}*:*${projectOwnerAliUid}*:project/*${projectName}*/logstore/*${logstoreName}*/consumergroup/*
log:ConsumerGroupUpdateCheckPoint	acs:log:*${regionName}*:*${projectOwnerAliUid}*:project/*${projectName}*/logstore/*${logstoreName}*/consumergroup/*${consumerGroupName}*
log:ConsumerGroupHeartBeat	acs:log:*${regionName}*:*${projectOwnerAliUid}*:project/*${projectName}*/logstore/*${logstoreName}*/consumergroup/*${consumerGroupName}*
log:UpdateConsumerGroup	acs:log:*${regionName}*:*${projectOwnerAliUid}*:project/*${projectName}*/logstore/*${logstoreName}*/consumergroup/*${consumerGroupName}*
log:GetConsumerGroupCheckPoint	acs:log:*${regionName}*:*${projectOwnerAliUid}*:project/*${projectName}*/logstore/*${logstoreName}*/consumergroup/*${consumerGroupName}*

For example, a project named project-test resides in the China (Hangzhou) region. The ID of the Apsara Stack tenant account to which the project belongs is 1234567. The name of the Logstore to be consumed is logstore-test and the consumer group name is consumergroup-test. To allow a RAM user to access the consumer group, you must grant the following permissions to the RAM user:

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "log:GetCursorOrData"
      ],
      "Resource": "acs:log:cn-hangzhou:1234567:project/project-test/logstore/logstore-test"
    },
    {
      "Effect": "Allow",
      "Action": [
        "log:CreateConsumerGroup",
        "log:ListConsumerGroup"
      ],
      "Resource": "acs:log:cn-hangzhou:1234567:project/project-test/logstore/logstore-test/consumergroup/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "log:ConsumerGroupUpdateCheckPoint",
        "log:ConsumerGroupHeartBeat",
        "log:UpdateConsumerGroup",
        "log:GetConsumerGroupCheckPoint"
      ],
      "Resource": "acs:log:cn-hangzhou:1234567:project/project-test/logstore/logstore-test/consumergroup/consumergroup-test"
    }
  ]
}
```

28.1.6.3.2. View the status of a consumer group

This topic describes how to use the console, API, and SDK to view the status of a consumer group. Log data consumption by using consumer groups is an advanced real-time data consumption method provided by Log Service. Automatic load balancing is implemented among multiple consumption instances. Spark Streaming and Storm use consumer groups as the basic mode to consume log data.

View the consumption progress in the console

1. Log on to the Log Service console.
2. In the Projects section, click the target project.
3. Find the target Logstore, and choose > **Data Consumption**.
4. Click the consumer group whose data consumption progress you want to view. The data consumption progress of each shard in the Logstore is displayed.

Use an API or SDK to view the data consumption progress

The following example uses the SDK for Java to describe how to call API operations to view the data consumption progress.

```
package test;
import java.util.ArrayList;
import com.aliyun.openservices.log.Client;
import com.aliyun.openservices.log.common.Consts.CursorMode;
import com.aliyun.openservices.log.common.ConsumerGroup;
import com.aliyun.openservices.log.common.ConsumerGroupShardCheckPoint;
import com.aliyun.openservices.log.exception.LogException;
public class ConsumerGroupTest {
    static String endpoint = "";
    static String project = "";
    static String logstore = "";
    static String accesskeyId = "";
    static String accesskey = "";
    public static void main(String[] args) throws LogException {
        Client client = new Client(endpoint, accesskeyId, accesskey);
        //Query all consumer groups of this Logstore. If no consumer group exists, the length of cons
umerGroups is 0.
        ArrayList<ConsumerGroup>  consumerGroups;
        try{
            consumerGroups = client.ListConsumerGroup(project, logstore).GetConsumerGroups();
        }
        catch(LogException e){
            if(e.GetErrorCode() == "LogStoreNotExist")
                System.out.println("this logstore does not have any consumer group");
            else{
                //internal server error branch
            }
            return;
        }
        for(ConsumerGroup c: consumerGroups){
            //Print consumer group properties, including the name, heartbeat timeout, and whether dat
a is consumed in order.
            System.out.println("Name: " + c.getConsumerGroupName());
            System.out.println("Heartbeat timeout: " + c.getTimeout());
            System.out.println("Consumption in order: " + c.isInOrder());
            for(ConsumerGroupShardCheckPoint cp: client.GetCheckPoint(project, logstore, c.getConsume
rGroupName()).GetCheckPoints()){
                System.out.println("shard: " + cp.getShard());
                //Format the returned time. The time is a long integer that is accurate to millisecon
ds.
                System.out.println("The last time when data was consumed: " + cp.getUpdateTime());
                System.out.println("Consumer name: " + cp.getConsumer());
                String consumerPrg = "";
                if(cp.getCheckPoint().isEmpty())
                    consumerPrg = "Consumption not started";
                else{
                    //The UNIX timestamp. Unit: seconds. Format the output value of the timestamp.
                    try{
                        int prg = client.GetPrevCursorTime(project, logstore, cp.getShard(), cp.getCh
eckPoint()).GetCursorTime();
                        consumerPrg = "" + prg;
                    }
                    catch(LogException e){
                        if(e.GetErrorCode() == "InvalidCursor")
                            consumerPrg = "Invalid. The previous consumption time has exceeded the re
tention period of the data in the Logstore.";
                        else{
```

```
                        else{
                            //internal server error
                            throw e;
                        }
                    }
                }
                System.out.println("Consumption progress: " + consumerPrg);
                String endCursor = client.GetCursor(project, logstore, cp.getShard(), CursorMode.END)
.GetCursor();
                int endPrg = 0;
                try{
                    endPrg = client.GetPrevCursorTime(project, logstore, cp.getShard(), endCursor).Ge
tCursorTime();
                }
                catch(LogException e){
                    //do nothing
                }
                //The UNIX timestamp. Unit: seconds. Format the output value of the timestamp.
                System.out.println("The time when the last data entry was received: " + endPrg);
            }
        }
    }
}
```

28.1.6.4. Use LogHub Storm to consume log data

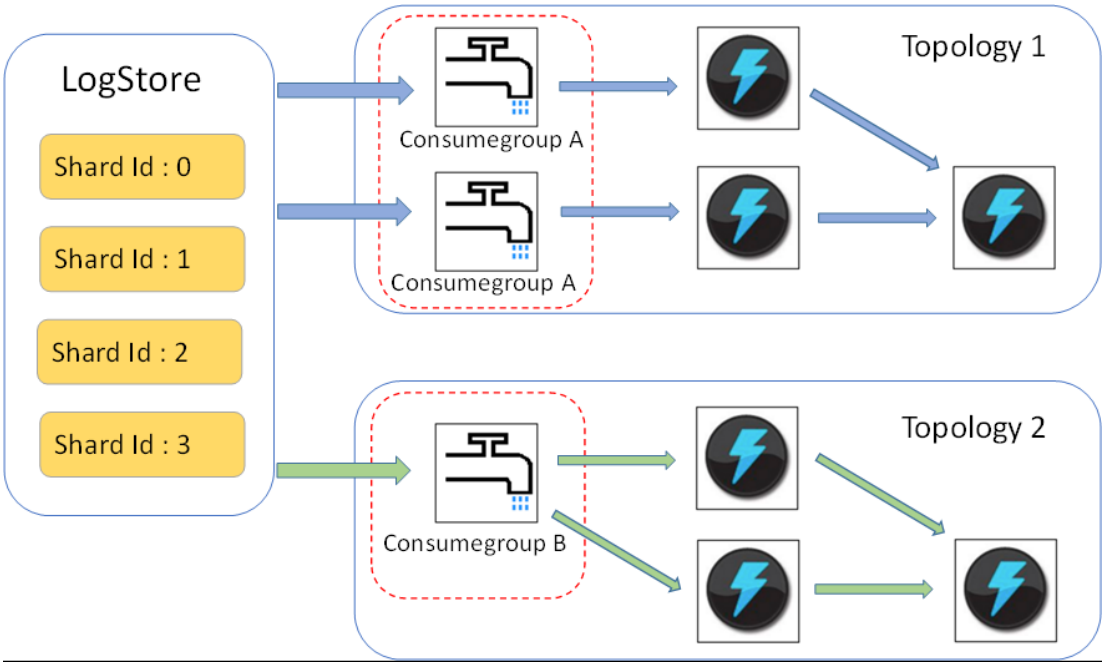
Log Service provides efficient and reliable log collection and consumption channels. You can use Logtail or the SDK to collect log data in real time. After logs are collected to LogHub, you can use Apache Spark Streaming, Apache Storm, and other real-time computation systems to consume the log data.

To reduce the consumption cost of LogHub data, Log Service provides LogHub Storm spouts for Storm users to read data from LogHub in real time.

Basic architecture and flowchart

- LogHub Storm spouts are enclosed in the boxes with red dashed lines. Each Storm topology has a group of spouts that read data from a Logstore. Spouts in different topologies are independent of each other.
- Each topology is identified by a unique LogHub consumer group name. Spouts in the same topology use a consumer library to achieve load balancing and automatic failover.
- Spouts read data from LogHub in real time, send data to bolts in the same topology, and then save consumption checkpoints to the LogHub server on a regular basis.

  Basic architecture and flowchart



Limits

- You can create up to 10 consumer groups to consume log data from a Logstore. If a consumer group is no longer in use, you can call the DeleteConsumerGroup operation of the SDK for Java to delete the consumer group.
- We recommend that the number of spouts be equal to the number of shards in a Logstore. This is because a single spout may be unable to process a large amount of data in multiple shards.
- If the data volume in a shard exceeds the processing capacity of a single spout, you can split the shard to reduce its data volume.
- LogHub spouts and bolts must use the ack method to check whether log data is successfully sent from spouts to bolts and whether the data is successfully processed by the bolts.

Examples

- Use the following code to create spouts and construct a topology:

```
     public static void main( String[] args )
    {
        String mode = "Local"; // Specify to use the local test mode.
           String conumser_group_name = ""; // Specify a unique consumer group name for each topol
ogy. The name cannot be an empty string. It must be 3 to 63 characters in length and can contain l
owercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase
letter or digit.
        String project = ""; // Specify the project in Log Service.
        String logstore = ""; // Specify the Logstore in Log Service.
        String endpoint = ""; // Specify the endpoint of Log Service.
        String access_id = ""; // Specify the AccessKey ID of the user.
        String access_key = "";
        // Configure a LogHub Storm spout.
        LogHubSpoutConfig config = new LogHubSpoutConfig(conumser_group_name,
                endpoint, project, logstore, access_id,
                access_key, LogHubCursorPosition.END_CURSOR);
        TopologyBuilder builder = new TopologyBuilder();
        // Create a LogHub Storm spout.
        LogHubSpout spout = new LogHubSpout(config);
        // You can create the same number of spouts as that of shards in a Logstore in actual busi
ness scenarios.
```

```
        builder.setSpout("spout", spout, 1);
        builder.setBolt("exclaim", new SampleBolt()).shuffleGrouping("spout");
        Config conf = new Config();
        conf.setDebug(false);
        conf.setMaxSpoutPending(1);
        // Configure the serialization method of LogGroupData by using the LogGroupDataSerializSer
ializer class if Kryo is used to serialize and deserialize data.
        Config.registerSerialization(conf, LogGroupData.class, LogGroupDataSerializSerializer.clas
s);
        if (mode.equals("Local")) {
            logger.info("Local mode...") ;
            LocalCluster cluster  = new LocalCluster();
            cluster.submitTopology("test-jstorm-spout", conf, builder.createTopology());
            try {
                Thread.sleep(6000 * 1000);    //waiting for several minutes
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            cluster.killTopology("test-jstorm-spout");
            cluster.shutdown();
        } else if (mode.equals("Remote")) {
            logger.info("Remote mode...");
            conf.setNumWorkers(2);
            try {
                StormSubmitter.submitTopology("stt-jstorm-spout-4", conf, builder.createTopology()
);
            } catch (AlreadyAliveException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (InvalidTopologyException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        } else {
            logger.error("invalid mode: " + mode);
        }
    }
}
```

- Use the following example code of bolts to consume log data and display the content of each log entry:

```
public class SampleBolt extends BaseRichBolt {
    private static final long serialVersionUID = 4752656887774402264L;
    private static final Logger logger = Logger.getLogger(BaseBasicBolt.class);
    private OutputCollector mCollector;
    @Override
    public void prepare(@SuppressWarnings("rawtypes") Map stormConf, TopologyContext context,
            OutputCollector collector) {
        mCollector = collector;
    }
    @Override
    public void execute(Tuple tuple) {
        String shardId = (String) tuple
                .getValueByField(LogHubSpout.FIELD_SHARD_ID);
        @SuppressWarnings("unchecked")
        List<LogGroupData> logGroupDatas = (ArrayList<LogGroupData>) tuple.getValueByField(LogHubS
pout.FIELD_LOGGROUPS);
        for (LogGroupData groupData : logGroupDatas) {
            // Each log group consists of one or more log entries.
            LogGroup logGroup = groupData.GetLogGroup();
            for (Log log : logGroup.getLogsList()) {
                StringBuilder sb = new StringBuilder();
                // Each log entry has a time field and other key-value pairs.
                int log_time = log.getTime();
                sb.append("LogTime:").append(log_time);
                for (Content content : log.getContentsList()) {
                    sb.append("\t").append(content.getKey()).append(":")
                            .append(content.getValue());
                }
                logger.info(sb.toString());
            }
        }
        // Spouts must use the ack method to indicate whether data has been successfully sent to b
olts.
        // In addition, bolts must use the ack method to indicate whether data is successfully pro
cessed by the bolts.
        mCollector.ack(tuple);
    }
    @Override
    public void declareOutputFields(OutputFieldsDeclarer declarer) {
        //do nothing
    }
}
```

Maven

Use the following code to add Maven dependencies for Storm versions earlier than Storm 1.0 (for example, Storm 0.9.6):

```
<dependency>
  <groupId>com.aliyun.openservices</groupId>
  <artifactId>loghub-storm-spout</artifactId>
  <version>0.6.6</version>
</dependency>
```

Use the following code to add Maven dependencies for Storm 1.0 and later versions:

```
<dependency>
  <groupId>com.aliyun.openservices</groupId>
  <artifactId>loghub-storm-1.0-spout</artifactId>
  <version>0.1.3</version>
</dependency>
```

28.1.6.5. Use Flume to consume log data

This topic describes how to use Flume to consume log data. You can use the aliyun-log-flume plug-in to connect LogHub of Log Service to Flume, and then write and consume log data.

The aliyun-log-flume plug-in connects LogHub to Flume. When LogHub is connected to Flume, you can connect Log Service to other systems such as HDFS and Kafka through Flume. The aliyun-log-flume plug-in provides the Sink and Source methods to connect Log Service to Flume.

- Sink: uses Flume to read data from other data sources and then writes data to LogHub.
- Source: uses Flume to consume LogHub data and then writes data to other systems.

LogHub Sink

You can use the Sink method to transmit data from other data sources to LogHub through Flume. Data can be parsed into the following two formats:

- SIMPLE: writes a Flume event to LogHub as a field.
- DELIMITED: delimits a Flume event with delimiters, parses an event into fields based on the configured column names, and then writes the fields to LogHub.

The following table lists the parameters you can configure when you use the Sink method to read data.

Parameter	Required	Description
type	Yes	Valid value: com.aliyun.loghub.flume.sink.LoghubSink.
endpoint	Yes	The endpoint of Log Service.
project	Yes	The name of the project.
logstore	Yes	The name of the Logstore.
accessKeyId	Yes	The AccessKey ID of your Apsara Stack tenant account.
accessKey	Yes	The AccessKey secret of your Apsara Stack tenant account.
batchSize	No	The number of log entries that are written to LogHub at a time. Default value: 1000.
maxBufferSize	No	The maximum size of the queue in the buffer. Default value: 1000.

Parameter	Required	Description
serializer	No	The serialization format of log data. Valid values: • **DELIMITED**: Data is parsed into the DELIMITED format. If you set this parameter to DELIMITED, you must set the columns parameter. • **SIMPLE**: Data is parsed into the SIMPLE format. This is the default value. • Custom **serializer**: Data is parsed into a custom serialization format. If you set this parameter to a custom serializer, you must specify the full name of the class.
columns	No	The configured column names. You must set this parameter if you set the serializer parameter to **DELIMITED**. Separate multiple columns with commas (,). Ensure that the columns are sorted in the same order as those of the log data.
separatorChar	No	The delimiter, which must be a single character. You can set this parameter if you set the serializer parameter to **DELIMITED**. Default value: `,`.
quoteChar	No	The quote character. You can set this parameter if you set the serializer parameter to **DELIMITED**. Default value: `""`.
escapeChar	No	The escape character. You can set this parameter if you set the serializer parameter to **DELIMITED**. Default value: `""`.
useRecordTime	No	Specifies whether to use the value of the timestamp field as the time when log data is written to Log Service. The default value false indicates that the current time is used.

Loghub Source

You can use the Source method to ship data from LogHub to other data systems through Flume. Data can be output in the following two formats:

- DELIMITED: writes delimited log data to Flume.
- JSON: writes JSON-formatted log data to Flume.

The following table lists the parameters you can configure when you use the Source method to read data.

Parameter	Required	Description
type	Yes	Valid value: com.aliyun.loghub.flume.source.LoghubSource.
endpoint	Yes	The endpoint of Log Service.
project	Yes	The name of the project.
logstore	Yes	The name of the Logstore.
accessKeyId	Yes	The AccessKey ID of your Apsara Stack tenant account.

Parameter	Required	Description
accessKey	Yes	The AccessKey secret of your Apsara Stack tenant account.
heartbeatIntervalMs	No	The heartbeat interval between the Flume client and LogHub. Unit: milliseconds. Default value: 30000.
fetchIntervalMs	No	The interval for pulling data from LogHub. Unit: milliseconds. Default value: 100.
fetchInOrder	No	Specifies whether to consume log data in the order that log data was generated. Default value: false.
batchSize	No	The number of log entries that are read at a time. Default value: 100.
consumerGroup	No	The name of the consumer group that reads data. The name is randomly generated.
initialPosition	No	The start point from which data is read. Valid values: begin, end, and timestamp. Default value: begin. **Note** If a checkpoint exists on the server, the checkpoint is used.
timestamp	No	The Unix timestamp. You must set this parameter if you set the initialPosition parameter to **timestamp**. Unix timestamp.
deserializer	Yes	The deserialization format of log data. Valid values: • **DELIMITED**: Data is parsed into the DELIMITED format. This is the default value. If you set this parameter to DELIMITED, you must set the columns parameter. • **JSON**: Data is parsed into the JSON format. • Custom **deserializer**: Data is parsed into a custom deserialization format. If you set this parameter to a custom deserializer, you must specify the full name of the class.
columns	No	The configured column names. You must set this parameter if you set the deserializer parameter to **DELIMITED**. Separate multiple columns with commas (,). Ensure that the columns are sorted in the same order as those of the log data.
separatorChar	No	The delimiter, which must be a single character. You can set this parameter if you set the deserializer parameter to **DELIMITED**. Default value: `,`.
quoteChar	No	The quote character. You can set this parameter if you set the deserializer parameter to **DELIMITED**. Default value: `""`.
escapeChar	No	The escape character. You can set this parameter if you set the deserializer parameter to **DELIMITED**. Default value: `""`.

Parameter	Required	Description
appendTimestamp	No	Specifies whether to append the timestamp as a field to the end of each log entry. You can set this parameter if you set the deserializer parameter to **DELIMITED**. Default value: false.
sourceAsField	No	Specifies whether to add the log source as a field named __source__. You can set this parameter if you set the deserializer parameter to **JSON**. Default value: false.
tagAsField	No	Specifies whether to add the log tags as a field with the field name __tag__: {tag names}. You can set this parameter if you set the deserializer parameter to **JSON**. Default value: false.
timeAsField	No	Specifies whether to add the log time as a field named __time__. You can set this parameter if you set the deserializer parameter to **JSON**. Default value: false.
useRecordTime	No	Specifies whether to use the value of the timestamp field as the time when log data is read from Log Service. The default value false indicates that the current time is used. Default value: false.

28.1.6.6. Use open source Flink to consume log data

Log Service provides the flink-log-connector plug-in to connect with Flink. This topic describes how to integrate the flink-log-connector plug-in with Flink to consume log data.

Prerequisites

- An AccessKey pair, a Log Service project, and a Logstore are created.
- If you log on to Log Service with a RAM user, relevant permissions to access a Logstore are granted to a RAM user. For more information, see Grant permissions to a RAM role.

Context

The flink-log-collector plug-in includes flink-log-consumer and flink-log-producer.

- The flink-log-consumer plug-in reads data from Log Service. This plug-in supports the exactly-once semantics and load balancing among shards.
- The flink-log-producer plug-in writes data into Log Service. When you use the flink-log-producer plug-in, you must add the following Maven dependencies to a project:

```
<dependency>
   <groupId>com.aliyun.openservices</groupId>
   <artifactId>flink-log-connector</artifactId>
   <version>0.1.13</version>
</dependency>
<dependency>
   <groupId>com.google.protobuf</groupId>
   <artifactId>protobuf-java</artifactId>
   <version>2.5.0</version>
</dependency>
```

Log Consumer

The flink-log-consumer plug-in can consume the log data of a Logstore in Log Service. The exactly-once semantics is achieved during log consumption. The flink-log-consumer plug-in detects the change of the number of shards in a Logstore. This increases efficiency.

Each Flink subtask consumes data of some shards in a Logstore. If shards in a Logstore are split or merged, the shards consumed by the subtask also change.

When you use the flink-log-consumer plug-in to consume data from Log Service, you can call the following API operations:

- GetCursorOrData

  You can call this operation to pull log data from a shard. Frequent API requests may exceed the read speed and IOPS limits of the shard. You can specify the ConfigConstants.LOG_FETCH_DATA_INTERVAL_MILLIS and ConfigConstants.LOG_MAX_NUMBER_PER_FETCH parameters to control the interval of API requests and number of log entries pulled in each request.

  ```
  configProps.put(ConfigConstants.LOG_FETCH_DATA_INTERVAL_MILLIS, "100");
  configProps.put(ConfigConstants.LOG_MAX_NUMBER_PER_FETCH, "100");
  ```

- ListShards

  You can call this operation to view all shards in a Logstore and the status of each shard. If the shards are frequently split and merged, you can adjust the call interval to detect the changes in the number of shards.

  ```
  // Call the ListShards operation once every 30 seconds.
  configProps.put(ConfigConstants.LOG_SHARDS_DISCOVERY_INTERVAL_MILLIS, "30000");
  ```

- CreateConsumerGroup

  You can call this operation to create a consumer group to synchronize checkpoints. This operation can be called only when consumption progress monitoring is enabled.

- ConsumerGroupUpdateCheckPoint

  You can call this operation to synchronize snapshots of Flink to a consumer group.

The following table lists the Apsara Stack resources required for RAM users to call the preceding API operations.

API	Alibaba Resource Name (ARN)
GetCursorOrData	`acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}`
ListShards	`acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}`
CreateConsumerGroup	`acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/*`
ConsumerGroupUpdateCheckPoint	`acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/${consumerGroupName}`

For more information, see Grant permissions to a RAM role.

1. Configure startup parameters.

   The following example shows how to consume log data. The java.util.Properties class is used as the configuration tool. You can find all constants to be configured in the ConfigConstants class.

```
Properties configProps = new Properties();
// Specify the endpoint of Log Service.
configProps.put(ConfigConstants.LOG_ENDPOINT, "cn-hangzhou.log.aliyuncs.com");
// Specify the AccessKey pair.
configProps.put(ConfigConstants.LOG_ACCESSSKEYID, "");
configProps.put(ConfigConstants.LOG_ACCESSKEY, "");
// Specify the project.
configProps.put(ConfigConstants.LOG_PROJECT, "ali-cn-hangzhou-sls-admin");
// Specify the Logstore.
configProps.put(ConfigConstants.LOG_LOGSTORE, "sls_consumergroup_log");
// Specify the start position to consume logs.
configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, Consts.LOG_END_CURSOR);
// Specify the data deserialization method.
RawLogGroupListDeserializer deserializer = new RawLogGroupListDeserializer();
final StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();
DataStream<RawLogGroupList> logTestStream = env.addSource(
        new FlinkLogConsumer<RawLogGroupList>(deserializer, configProps));
```

Note The number of subtasks in the Flink Streaming is independent of the number of shards in a Logstore. If the number of shards is greater than that of subtasks, each subtask consumes multiple shards exactly once. If the number of shards is less than that of subtasks, some subtasks are idle until new shards are generated.

2. Specify the start consumption position.

   The flink-log-consumer plug-in enables you to specify the start consumption position of log data in a shard. By specifying the ConfigConstants.LOG_CONSUMER_BEGIN_POSITION parameter, you can start data consumption from the earliest, latest, or a specific time point. The flink-log-connector plug-in also allows a consumer group to resume consumption from a specific position. Valid values:

   - Consts.LOG_BEGIN_CURSOR: The consumption starts from the earliest data.
   - Consts.LOG_END_CURSOR: The consumption starts from the latest data.
   - Consts.LOG_FROM_CHECKPOINT: The consumption starts from a checkpoint that is stored in a specific consumer group. You can use the ConfigConstants.LOG_CONSUMERGROUP parameter to specify the consumer group.
   - UnixTimestamp: a string of the INTEGER data type. The timestamp is the number of seconds that have elapsed since 00:00:00 January 1, 1970. The value indicates that data in a shard is consumed from this time point.

   You can use the following code to specify a start consumption position:

   ```
   configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, Consts.LOG_BEGIN_CURSOR);
   configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, Consts.LOG_END_CURSOR);
   configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, "1512439000");
   configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, Consts.LOG_FROM_CHECKPOINT);
   ```

   Note If you have configured consumption resumption from a state backend of Flink when you start a Flink job, the flink-log-connector plug-in uses checkpoints stored in the state backend.

3. (Optional)Configure consumption progress monitoring.

   The flink-log-consumer plug-in enables you to configure consumption progress monitoring. Consumption progress indicates the real-time consumption position of each shard. These positions are indicated by timestamps. For more information, see View the status of a consumer group.

   ```
   configProps.put(ConfigConstants.LOG_CONSUMERGROUP, "your consumer group name");
   ```

> **Note** This configuration item is optional. If you specify this configuration item and no consumer group exists, the flink-log-consumer plug-in creates a consumer group. Snapshots in the flink-log-consumer plug-in are automatically synchronized to the consumer group of Log Service, and you can view the consumption progress of the flink-log-consumer plug-in in the Log Service console.

4. Configure consumption resumption and the exactly-once semantics.

   If the checkpointing feature of Flink is enabled, the flink-log-consumer plug-in periodically stores the consumption progress of each shard. When a job fails, Flink restores the flink-log-consumer plug-in and starts to consume data from the latest checkpoint.

   While you configure the checkpointing period, the maximum amount of data to be re-consumed when a failure occurs is defined. You can use the following code to configure the checkpointing period:

   ```
   final StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();
   // Configure the exactly-once semantics.
   env.getCheckpointConfig().setCheckpointingMode(CheckpointingMode.EXACTLY_ONCE);
   // Store checkpoints every five seconds.
   env.enableCheckpointing(5000);
   ```

   For more information about the Flink checkpoints, see Checkpoints in the Flink documentation.

Log Producer

The flink-log-producer plug-in writes data into Log Service.

> **Note** The flink-log-producer plug-in supports the Flink at-least-once semantics. If a job fails, data written into Log Service may be duplicated but never lost.

When you use the flink-log-producer plug-in to writes data to Log Service, you can call the following API operations:

- PostLogStoreLogs
- ListShards

The following table lists the Apsara Stack resources required for RAM users to call the preceding API operations.

API	ARN
PostLogStoreLogs	`acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}`
ListShards	`acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}`

For more information about RAM users and how to authorize RAM users, see Grant permissions to a RAM role.

1. Initialize the flink-log-producer plug-in.

i. Configure startup parameters for the flink-log-producer plug-in.

The initialization process for the flink-log-producer plug-in is similar to that for the flink-log-consumer plug-in. The following code shows the available parameters that you can configure for the flink-log-producer plug-in. You can use the default values of these parameters. You can also specify the parameters based on your needs.

```
// The number of I/O threads used to send data. Default value: 8.
ConfigConstants.LOG_SENDER_IO_THREAD_COUNT
// The time it takes to send the data after log data is cached. Default value: 3000.
ConfigConstants.LOG_PACKAGE_TIMEOUT_MILLIS
// The number of logs in the cached package. Default value: 4096.
ConfigConstants.LOG_LOGS_COUNT_PER_PACKAGE
// The size of the cached package. Default value: 3 Mbits.
ConfigConstants.LOG_LOGS_BYTES_PER_PACKAGE
// The total memory size that the job can use. Default value: 100 Mbits.
ConfigConstants.LOG_MEM_POOL_BYTES
```

> **Note** These parameters are optional. You can use their default values.

ii. Reload LogSerializationSchema to define the method of serializing data into RawLogGroup.

To use the hash key to specify the shard for data writes, you can use the LogPartitioner method to generate the hash key for the data.

Example:

```
FlinkLogProducer<String> logProducer = new FlinkLogProducer<String>(new SimpleLogSerializer()
, configProps);
logProducer.setCustomPartitioner(new LogPartitioner<String>() {
      // Generate a 32-bit hash value.
      public String getHashKey(String element) {
          try {
              MessageDigest md = MessageDigest.getInstance("MD5");
              md.update(element.getBytes());
              String hash = new BigInteger(1, md.digest()).toString(16);
              while(hash.length() < 32) hash = "0" + hash;
              return hash;
          } catch (NoSuchAlgorithmException e) {
          }
          return  "00000000000000000000000000000000000000000000000000000000000000000000";
      }
  });
```

> **Note** The LogPartitioner method is optional. If you do not configure this method, data is randomly written into a shard.

2. Run the following code and write the generated strings to Log Service.

```
// Serialize data into the format of raw log groups.
class SimpleLogSerializer implements LogSerializationSchema<String> {
    public RawLogGroup serialize(String element) {
        RawLogGroup rlg = new RawLogGroup();
        RawLog rl = new RawLog();
        rl.setTime((int)(System.currentTimeMillis() / 1000));
        rl.addContent("message", element);
        rlg.addLog(rl);
        return rlg;
    }
}
public class ProducerSample {
    public static String sEndpoint = "cn-hangzhou.log.aliyuncs.com";
    public static String sAccessKeyId = "";
    public static String sAccessKey = "";
    public static String sProject = "ali-cn-hangzhou-sls-admin";
    public static String sLogstore = "test-flink-producer";
    private static final Logger LOG = LoggerFactory.getLogger(ConsumerSample.class);
    public static void main(String[] args) throws Exception {
        final ParameterTool params = ParameterTool.fromArgs(args);
        final StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment
();
        env.getConfig().setGlobalJobParameters(params);
        env.setParallelism(3);
        DataStream<String> simpleStringStream = env.addSource(new EventsGenerator());
        Properties configProps = new Properties();
        // Specify the endpoint of Log Service.
        configProps.put(ConfigConstants.LOG_ENDPOINT, sEndpoint);
        // Specify the AccessKey pair to access Log Service.
        configProps.put(ConfigConstants.LOG_ACCESSSKEYID, sAccessKeyId);
        configProps.put(ConfigConstants.LOG_ACCESSKEY, sAccessKey);
        // Specify the project to which logs are written.
        configProps.put(ConfigConstants.LOG_PROJECT, sProject);
        // Specify the Logstore to which logs are written.
        configProps.put(ConfigConstants.LOG_LOGSTORE, sLogstore);
        FlinkLogProducer<String> logProducer = new FlinkLogProducer<String>(new SimpleLogSerializ
er(), configProps);
        simpleStringStream.addSink(logProducer);
        env.execute("flink log producer");
    }
    // Simulate log generation.
    public static class EventsGenerator implements SourceFunction<String> {
        private boolean running = true;
        @Override
        public void run(SourceContext<String> ctx) throws Exception {
            long seq = 0;
            while (running) {
                Thread.sleep(10);
                ctx.collect((seq++) + "-" + RandomStringUtils.randomAlphabetic(12));
            }
        }
        @Override
        public void cancel() {
            running = false;
        }
    }
}
```

28.1.6.7. Use Logstash to consume log data

Log Service allows you to use SDKs developed in various languages and various stream computing systems to consume log data. In addition, Log Service allows you to use Logstash to consume log data. You can configure the Logstash input plug-in to read log data from Log Service and write the data to other systems, such as Kafka and Hadoop Distributed File System (HDFS).

Features

- **Distributed collaborative consumption**: Multiple servers can be configured to consume log data from a Logstore at the same time.
- **High performance**: If you use consumer groups written in Java to consume log data, the consumption speed of a CPU core can reach up to 20 MB/s.
- **High reliability**: Log Service saves the consumption progress. This mechanism enables automatic resumption of log consumption from the last checkpoint after a consumption exception is solved.
- **Automatic load balancing**: Shards are automatically allocated based on the number of consumers in a consumer group. The shards are reallocated if consumers join or leave the consumer group.

28.1.6.8. Use Spark Streaming to consume log data

This topic describes how to use Spark Streaming to consume log data. After logs are collected to Log Service, you can use the Spark SDK provided by Log Service to process log data in Spark Streaming.

The Spark SDK supports two consumption modes: Receiver and Direct.

The Maven dependency is as follows:

```
<dependency>
  <groupId>com.aliyun.emr</groupId>
  <artifactId>emr-logservice_2.11</artifactId>
  <version>1.7.2</version>
</dependency>
```

Receiver mode

In the Receiver mode, a consumer group consumes data from Log Service and temporarily stores the data in Spark Executor. After a Spark Streaming job is started, it reads and processes data from Spark Executor. For more information, see Use consumer groups to consume log data. Each log entry is returned as a JSON string. The consumer group periodically saves checkpoints to Log Service. You do not need to update checkpoints.

- Example code

```
import org.apache.spark.storage.StorageLevel
import org.apache.spark.streaming.aliyun.logservice.LoghubUtils
import org.apache.spark.streaming.{ Milliseconds, StreamingContext}
import org.apache.spark.SparkConf
object TestLoghub {
  def main(args: Array[String]): Unit = {
    if (args.length < 7) {
      System.err.println(
        """Usage: TestLoghub <project> <logstore> <loghub group name> <endpoint>
          |         <access key id> <access key secret> <batch interval seconds>
        """.stripMargin)
      System.exit(1)
    }
    val project = args(0)
    val logstore = args(1)
    val consumerGroup = args(2)
    val endpoint = args(3)
    val accessKeyId = args(4)
    val accessKeySecret = args(5)
    val batchInterval = Milliseconds(args(6).toInt * 1000)
    def functionToCreateContext(): StreamingContext = {
      val conf = new SparkConf().setAppName("Test Loghub")
      val ssc = new StreamingContext(conf, batchInterval)
      val loghubStream = LoghubUtils.createStream(
        ssc,
        project,
        logstore,
        consumerGroup,
        endpoint,
        accessKeyId,
        accessKeySecret,
        StorageLevel.MEMORY_AND_DISK)
      loghubStream.checkpoint(batchInterval * 2).foreachRDD(rdd =>
        rdd.map(bytes => new String(bytes)).top(10).foreach(println)
      )
      ssc.checkpoint("hdfs:///tmp/spark/streaming") // set checkpoint directory
      ssc
    }
    val ssc = StreamingContext.getOrCreate("hdfs:///tmp/spark/streaming", functionToCreateContext
_)
    ssc.start()
    ssc.awaitTermination()
  }
}
```

- Parameter description

Parameter	Type	Description
project	String	The project in Log Service.
logstore	String	The Logstore in Log Service.
consumerGroup	String	The name of the consumer group.
endpoint	String	The endpoint of the region to which the project belongs.

Parameter	Type	Description
accessKeyId	String	The AccessKey ID used to access Log Service.
accessKeySecret	String	The AccessKey secret used to access Log Service.

- Notes

  In the Receiver mode, data loss may occur in some cases. To avoid data loss, you can turn on the Write-Ahead Logs switch, which is supported in Spark 1.2 and later versions. For more information, visit Spark Streaming Programming Guide.

Direct mode

In the Direct mode, no consumer group is required. API operations are called to request data from Log Service. Compared with the Receiver mode, the Direct mode has the following benefits:

- Simplified parallelism. The number of Spark partitions is the same as the number of shards in a Logstore. You can split shards to improve the parallelism of tasks.
- Increased efficiency. You no longer need to turn on the Write-Ahead Logs switch to prevent data loss.
- Exactly-once semantics. Data is read directly from Log Service. Checkpoints are submitted after the task is successful. In some cases, data may be repeatedly consumed if the task is not ended due to unexpected exit of Spark.

You must configure the ZooKeeper service when you use the Direct mode. You must set a checkpoint directory in ZooKeeper to store intermediate state data. If you want to re-consume data after restarting a task, you must delete the directory from ZooKeeper and change the name of the consumer group.

- Example code

```
import com.aliyun.openservices.loghub.client.config.LogHubCursorPosition
import org.apache.spark.SparkConf
import org.apache.spark.streaming.{ Milliseconds, StreamingContext}
import org.apache.spark.streaming.aliyun.logservice.{ CanCommitOffsets, LoghubUtils}
object TestDirectLoghub {
  def main(args: Array[String]): Unit = {
    if (args.length < 7) {
      System.err.println(
        """Usage: TestDirectLoghub <project> <logstore> <loghub group name> <endpoint>
          |         <access key id> <access key secret> <batch interval seconds> <zookeeper host:p
ort=localhost:2181>
        """.stripMargin)
      System.exit(1)
    }
    val project = args(0)
    val logstore = args(1)
    val consumerGroup = args(2)
    val endpoint = args(3)
    val accessKeyId = args(4)
    val accessKeySecret = args(5)
    val batchInterval = Milliseconds(args(6).toInt * 1000)
    val zkAddress = if (args.length >= 8) args(7) else "localhost:2181"
    def functionToCreateContext(): StreamingContext = {
      val conf = new SparkConf().setAppName("Test Direct Loghub")
      val ssc = new StreamingContext(conf, batchInterval)
      val zkParas = Map("zookeeper.connect" -> zkAddress,
        "enable.auto.commit" -> "false")
      val loghubStream = LoghubUtils.createDirectStream(
        ssc,
        project,
        logStore,
        consumerGroup,
        accessKeyId,
        accessKeySecret,
        endpoint,
        zkParas,
        LogHubCursorPosition.END_CURSOR)
      loghubStream.checkpoint(batchInterval).foreachRDD(rdd => {
        println(s"count by key: ${rdd.map(s => {
          s.sorted
          (s.length, s)
        }).countByKey().size}")
        loghubStream.asInstanceOf[CanCommitOffsets].commitAsync()
      })
      ssc.checkpoint("hdfs:///tmp/spark/streaming") // set checkpoint directory
      ssc
    }
    val ssc = StreamingContext.getOrCreate("hdfs:///tmp/spark/streaming", functionToCreateContext
_)
    ssc.start()
    ssc.awaitTermination()
  }
}
```

- Parameter description

Parameter	Type	Description
project	String	The project in Log Service.
logstore	String	The Logstore in Log Service.
consumerGroup	String	The name of the consumer group (only used to save consumption checkpoints).
endpoint	String	The endpoint of the region to which the project belongs.
accessKeyId	String	The AccessKey ID used to access Log Service.
accessKeySecret	String	The AccessKey secret used to access Log Service.
zkAddress	String	The endpoint of ZooKeeper.

- Parameter settings

  In the Direct mode, you must specify the number of log entries that are consumed in each shard in each batch. Otherwise, the data reading process cannot be ended. You can throttle the transmission rate of a single batch by setting the two parameters listed in the following table.

Parameter	Description	Default value
spark.loghub.batchGet.step	The number of log groups returned for a single request.	100
spark.streaming.loghub.maxRatePerShard	The maximum number of log entries that are read in a shard.	10,000

  The number of log entries processed in each batch is calculated as follows: `number of shards x max(spark.loghub.batchGet.step x n x number of log entries in a log group, spark.streaming.loghub.maxRatePerShard x duration)`.

  - n: the number of requests required to increase the returned rows to `spark.streaming.loghub.maxRatePerShard x duration`.
  - duration: the interval between batch processing. Unit: milliseconds.

  If you need to combine multiple data streams, the number of shards refers to the total number of shards in all Logstores.

  - Example

    For example, the number of shards is 100. Each log group contains 50 log entries. Batches are processed at an interval of two seconds. If you want to process 20,000 log entries in each batch, use the following configurations:

    - `spark.loghub.batchGet.step: 4`
    - `spark.streaming.loghub.maxRatePerShard: 100`

    If each log group contains 60 log entries and you want to process 20,000 log entries in each batch, 24,000 log entries will be processed based on the preceding configurations (`60 x 4 x 100 = 24,000`).

- Accurate transmission rate throttling

  A smaller `spark.loghub.batchGet.step` value increases the accuracy of throttling and the number of requests. We recommend that you count the average number of log entries contained in a log group and then set the preceding two parameters.

28.1.6.9. Use Realtime Compute to consume log data

You can use Realtime Compute (Blink) to create a schema for Log Service data and consume the data.

Log Service stores streaming data. Therefore, Realtime Compute can use the streaming data as input data. In Log Service, each log entry contains multiple fields and each field is a key-value pair. The following example is a sample log entry:

```
__source__:  11.85.123.199
__tag__:__receive_time__:  1562125591
__topic__:  test-topic
a:  1234
b:  0
c:  hello
```

You can use the following data definition language (DDL) statement to create a table in Realtime Compute:

```
create table sls_stream(
  a int,
  b int,
  c varchar
) with (
  type ='sls',
  endPoint ='<your endpoint>',
  accessId ='<your access key id>',
  accessKey ='<your access key>',
  startTime = '2017-07-05 00:00:00',
  project ='ali-cloud-streamtest',
  logStore ='stream-test',
  consumerGroup ='consumerGroupTest1'
);
```

Attribute fields

Realtime Compute can extract fields from log content. In addition, Realtime Compute can extract three attribute fields and custom tag fields, such as `__receive_time__`. The following table lists the three attribute fields.

Attribute fields

Field name	Description
`__source__`	The source of the log entry.
`__topic__`	The topic of the log entry.
`__timestamp__`	The time when the log entry is generated.

To extract the preceding fields, you must add HEADERs in the DDL statement. For example:

```
create table sls_stream(
  __timestamp__ bigint HEADER,
  __receive_time__ bigint HEADER
  b int,
  c varchar
) with (
  type ='sls',
  endPoint ='<your endpoint>',
  accessId ='<your access key id>',
  accessKey ='<your access key>',
  startTime = '2017-07-05 00:00:00',
  project ='ali-cloud-streamtest',
  logStore ='stream-test',
  consumerGroup ='consumerGroupTest1'
);
```

Parameters in the WITH clause

The following table describes the parameters in the WITH clause.

Parameter	Required	Description
endPoint	Yes	The endpoint of Log Service.
accessId	Yes	The AccessKey ID of the Apsara Stack tenant account or RAM user that is used to access Log Service.
accessKey	Yes	The AccessKey secret of the Apsara Stack tenant account or RAM user that is used to access Log Service.
project	Yes	The name of the project in Log Service.
logStore	Yes	The name of the Logstore in Log Service.
consumerGroup	No	The name of the consumer group.
startTime	No	The time when Realtime Compute starts to consume data.
heartBeatIntervalMills	No	The heartbeat interval of the client that consumes log data. Unit: seconds. Default value: 10.
maxRetryTimes	No	The maximum number of retries to read data. Default value: 5.
batchGetSize	No	The number of log groups that are read at a time. Default value: 10. If the version of Blink is 1.4.2 or later, the default value is 100 and the maximum value is 1000.

Parameter	Required	Description
columnErrorDebug	No	Specifies whether to enable debugging. If debugging is enabled, log entries that fail to be parsed are displayed. Default value: `false`.

Field type mapping

All log fields in Log Service are of the string type. The following table lists the mapping between the type of Log Service fields and the type of Realtime Compute fields. We recommend that you declare the mapping in a data definition language (DDL) statement.

Data type of Log Service	Data type of Realtime Compute
STRING	VARCHAR

If you specify another data type to convert Log Service data, an automatic conversion attempt is performed. For example, you can specify *BIGINT* as the data type to convert the string "1000" and specify *timestamp* as the data type to convert the string "2018-01-12 12:00:00".

> **Note**
> - Blink versions earlier than 2.2.0 do not support shard scaling. If you split or merge shards when a job is reading data from a Logstore, the job fails and cannot continue. In this case, you must restart the job.
> - No versions of Blink allow you to delete or recreate a Logstore whose log data is being consumed.
> - For Blink version 1.6.0 and earlier, if you specify *consumerGroup* to consume log data from a Logstore that contains a large number of shards, the read performance may be affected.
> - When you create a schema, Log Service data cannot be converted to data of the map type.
> - Fields whose values are empty are set to *null*.
> - Unordered field conversions are supported. However, we recommend that you convert the fields in the same order as the fields in the schema.
> - The *batchGetSize* parameter specifies the number of log groups that are obtained based on the *logGroup* parameter. If the size of each log entry and the value of the *batchGetSize* parameter are both large, garbage collection (GC) of data in the memory may frequently occur.

Precautions

- If no new data is written to a shard, the overall latency of a job increases. In this case, you need to modify the number of concurrent tasks in the job to the number of shards from which data is read and written.
- We recommend that you set the number of concurrent tasks in a job to the same as the number of shards. Otherwise, data may be filtered out when the job reads historical data from two shards at significantly different speeds.
- To extract fields in tags such as `__tag__:__hostname__` and `__tag__:__path__`, you can delete the __tag__: prefix and use the method of extracting attribute fields.

  > **Note** This type of data cannot be extracted during debugging. We recommend that you use the local debugging method and the print method to display data in logs.

28.1.7. RAM

28.1.7.1. Overview

Resource Access Management (RAM) is a resource access control service provided by Apsara Stack.

You can use RAM to manage users, including employees, systems, and applications. You can also use RAM to grant users permissions to access resources.

RAM provides the following features:

- RAM Role

  To authorize a cloud service in a level-1 organization to use other resources in the organization, you must create a RAM role. This role specifies the operations that the cloud service can perform on the resources.

  Only system administrators and level-1 organization administrators can create RAM roles.

- User group

  You can create multiple RAM users for an organization and grant the users different permissions on the same cloud resources in the organization.

  You can create RAM user groups to classify and authorize RAM users within your Apsara Stack tenant account. This simplifies the management of RAM users and their permissions.

  You can create RAM policies to grant permissions to different user groups.

28.1.7.2. Create a RAM role

To authorize a cloud service in a level-1 organization to use other resources in the organization, you must create a RAM role. This role specifies the operations that the cloud service can perform on the resources.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.

   For more information, see Log on to the Log Service console.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the upper-right corner of the page, click **Create RAM Role**.
5. On the **Roles - Create RAM Role** page that appears, set the **Role Name** and **Description** parameters.
6. Click **Create**.

28.1.7.3. Create a user

This topic describes how an administrator creates a user and assigns a role to the user. The role varies based on the cloud resources that the user needs to access.

Procedure

1.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. On the **Users** page, click **Create**.
5. In the dialog box that appears, set the parameters based on your requirements.

Parameter	Description
Username	The username.
Display Name	The display name of the user.

Parameter	Description
Roles	The role that you want to assign to the user.
Organization	The organization to which the user belongs.
Logon Policy	The logon policy that restricts the logon time and IP address of the user. If you do not specify this parameter, the default policy is attached to the created user.
Mobile Number	The mobile phone number of the user. If you need to send text messages about the usage and requests for resources to the mobile phone number, make sure that the specified mobile phone number is correct.
Landline Number	Optional. The landline number of the user.
Email	The email address of the user. If you need to send emails about the usage and requests for resources to the email address, make sure that the specified email address is correct.
DingTalk Key	Optional. The DingTalk key.
Notify User by SMS	Specifies whether to send text messages about the usage and requests for resources to the specified mobile phone number.
Notify User by Email	Specifies whether to send emails about the usage and requests for resources to the specified email address.
Notify User by DingTalk	Specifies whether to send messages about the usage and requests for resources to the specified DingTalk user.

6. Click **OK**.

28.1.7.4. Create a RAM user group

This topic describes how to create a RAM user group in an organization and grant permissions to RAM users in the RAM user group.

Prerequisites

An organization is created.

Context

The relationships between RAM user groups and RAM users:

- A RAM user group can contain zero or more RAM users.
- A RAM user does not need to belong to a RAM user group.
- You can add a RAM user to multiple RAM user groups.

The relationships between RAM user groups and organizations:

- A RAM user group belongs to only one organization.
- You can create multiple RAM user groups in an organization.

The relationships between RAM user groups and RAM roles:

- Only one RAM role can be assigned to each RAM user group.
- A RAM role can be assigned to multiple RAM user groups.
- When a RAM role is assigned to a RAM user group, the permissions that the RAM role has are automatically

granted to RAM users in the RAM user group.

The relationships between RAM user groups and resource sets:

- You can add zero or more user groups to a resource set.
- A user group can be added to multiple resource sets.

Procedure

1.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **User Groups**.
4. In the upper-right corner of the page, click **Create User Group**.
5. In the dialog box that appears, set the **User Group Name** and **Organization** parameters.
6. Click **OK**.

28.1.7.5. Add a RAM user to a RAM user group

This topic describes how to add a RAM user to a RAM user group.

Procedure

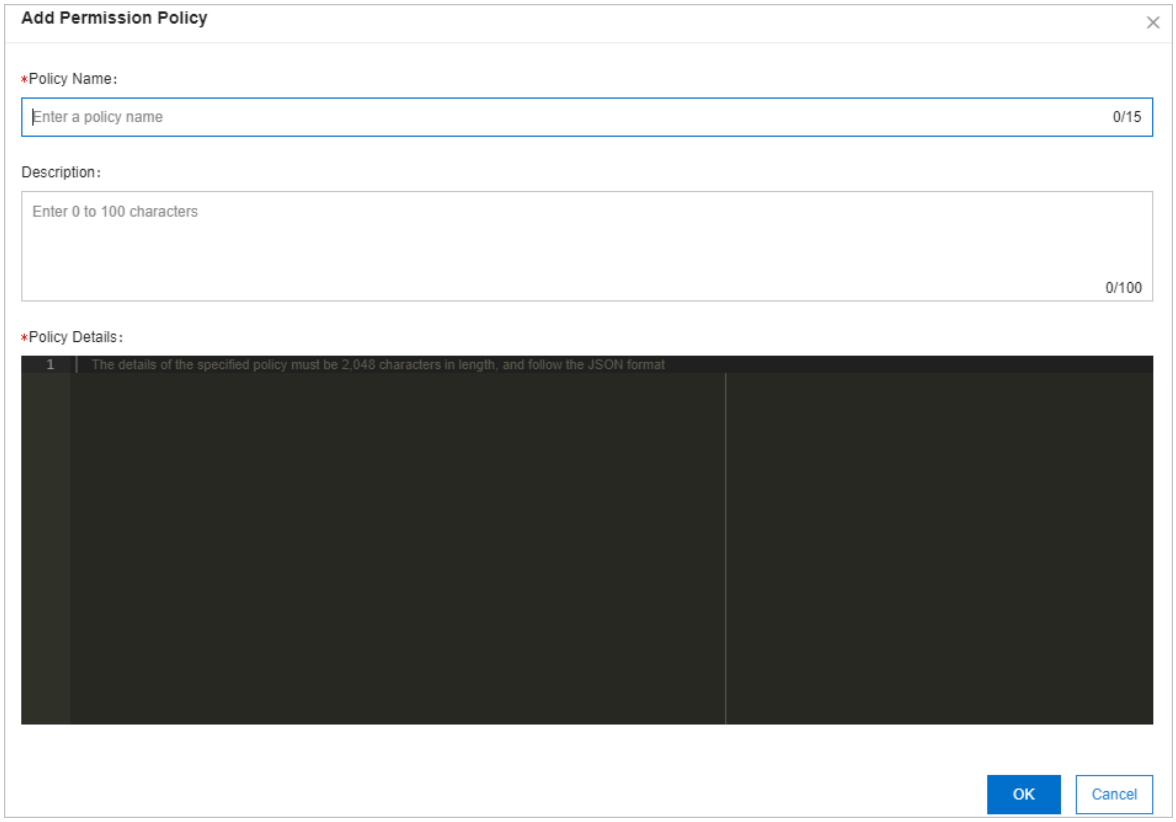
1.
2. In the top navigation bar, click **Enterprise**.
3.
4. Find the user group to which you want to add a RAM user, and click **Add User** in the **Actions** column.
5. In the dialog box that appears, select a RAM user from the left pane, and click the right arrow to move the RAM user to the right pane.
6. Click **OK**.

28.1.7.6. Create a permission policy

If you want to use a cloud service to access the resources of other cloud services, you must create a permission policy for a RAM role. Then, the policy is automatically attached to the RAM user group to which the RAM role is assigned.

Procedure

1.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the list of role names, find the RAM role for which you want to create a permission policy and choose **More > Modify**.
5. Click **Permissions**.
6. Click **Add Permission Policy**.
7. In the **Add Permission Policy** dialog box, enter the policy information.



For more information about how to specify the policy information, see Use custom policies to grant RAM user the required permissions.

28.1.7.7. Grant permissions to a RAM role

This topic describes how to grant permissions to a RAM role. After a RAM role is granted permissions, the RAM users in the associated RAM user group inherit the permissions.

Procedure

1.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4.
5. On the **Permissions** tab, click **Select Existing Permission Policy**.
6. In the **Select Existing Permission Policy** dialog box, select a permission policy and click **OK**.

   If no policies are available, create a policy. For more information, see Create a permission policy.

28.1.7.8. Use custom policies to grant RAM user the required permissions

This topic describes how to use custom Resource Access Management (RAM) policies to grant RAM users the required permissions. You can grant permissions to the RAM users under your Apsara Stack tenant account.

Context

For data security, we recommend that you follow the principle of least privilege (PoLP) when you grant permissions to RAM users. You must grant the read-only permission on the project list to RAM users. Otherwise, the RAM users cannot view the projects in the project list.

Use the RAM console to grant permissions to a RAM user

- The read-only permission on projects

  For example, you can use your Apsara Stack tenant account to grant RAM users the following permissions:

  - The permission to view the list of projects that belong to the Apsara Stack tenant account
  - The permission to read specific projects

  Use the following policy:

```
{
   "Version": "1",
   "Statement": [
     {
       "Action": ["log:ListProject"],
       "Resource": ["acs:log:*:*:project/*"],
       "Effect": "Alow"
      },
     {
       "Action": [
         "log:Get*",
         "log:List*"
       ],
       "Resource": "acs:log:*:*:project/<The name of the project>/*",
       "Effect": "Allow"
     }
   ]
 }
```

- The permission to read a Logstore, save searches, and use saved searches.

  For example, you can use your Apsara Stack tenant account to grant RAM users the following permissions:

  - The permission to view the project list of the Apsara Stack tenant account
  - The permission to read a Logstore, save searches, and use saved searches

  Use the following policy:

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "log:ListProject"
      ],
      "Resource": "acs:log:*:*:project/*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:List*"
      ],
      "Resource": "acs:log:*:*:project/<The name of the project>/logstore/*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:Get*",
        "log:List*"
      ],
      "Resource": [
        "acs:log:*:*:project/<The name of the project>/logstore/<The name of the Logstore>"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:List*"
      ],
      "Resource": [
        "acs:log:*:*:project/<The name of the project>/dashboard",
        "acs:log:*:*:project/<The name of the project>/dashboard/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:Get*",
        "log:List*",
        "log:Create*"
      ],
      "Resource": [
        "acs:log:*:*:project/<The name of the project>/savedsearch",
        "acs:log:*:*:project/<The name of the project>/savedsearch/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Note In the policy, a value of the Resource attribute that does not end with an asterisk (*) indicates the exact resource. A value that ends with an asterisk (*) indicates all resources that match the value.

- The permission to read a Logstore and view all saved searches and dashboards in a project

  For example, you can use your Apsara Stack tenant account to grant a RAM user the following permissions:

- The permission to view the project list of the Apsara Stack tenant account
- The permission to read a Logstore and view all saved searches and dashboards in a project

Use the following policy:

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "log:ListProject"
      ],
      "Resource": "acs:log:*:*:project/*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:List*"
      ],
      "Resource": "acs:log:*:*:project/<The name of the project>/logstore/*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:Get*",
        "log:List*"
      ],
      "Resource": [
        "acs:log:*:*:project/<The name of the project>/logstore/<The name of the Logstore>"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:Get*",
        "log:List*"
      ],
      "Resource": [
        "acs:log:*:*:project/<The name of the project>/dashboard",
        "acs:log:*:*:project/<The name of the project>/dashboard/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:Get*",
        "log:List*"
      ],
      "Resource": [
        "acs:log:*:*:project/<The name of the project>/savedsearch",
        "acs:log:*:*:project/<The name of the project>/savedsearch/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Grant RAM users the permissions that are required to call Log Service operations

- The permission to write data to a project

  To grant RAM users only the permission to write data to a project, use the following policy:

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "log:Post*"
      ],
      "Resource": "acs:log:*:*:project/<The name of the project>/*",
      "Effect": "Allow"
    }
  ]
}
```

- The permission to consume data of a project

  To grant RAM users only the permission to consume data of a project, use the following policy:

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "log:ListShards",
        "log:GetCursorOrData",
        "log:GetConsumerGroupCheckPoint",
        "log:UpdateConsumerGroup",
        "log:ConsumerGroupHeartBeat",
        "log:ConsumerGroupUpdateCheckPoint",
        "log:ListConsumerGroup",
        "log:CreateConsumerGroup"
      ],
      "Resource": "acs:log:*:*:project/<The name of the project>/*",
      "Effect": "Allow"
    }
  ]
}
```

- The permission to consume data of a Logstore

  To grant RAM users only the permission to consume data of a Logstore, use the following policy:

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "log:ListShards",
        "log:GetCursorOrData",
        "log:GetConsumerGroupCheckPoint",
        "log:UpdateConsumerGroup",
        "log:ConsumerGroupHeartBeat",
        "log:ConsumerGroupUpdateCheckPoint",
        "log:ListConsumerGroup",
        "log:CreateConsumerGroup"
      ],
      "Resource": [
        "acs:log:*:*:project/<The name of the project>/logstore/<The name of the Logstore>",
        "acs:log:*:*:project/<the name of the project>/logstore/<the name of the Logstore>/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

28.1.8. FAQ

28.1.8.1. Log collection

28.1.8.1.1. How do I troubleshoot Logtail collection errors?

If the Logtail preview page is blank or "No Data" is displayed on the query page, perform the following steps:

Procedure

1. Check whether Log Service receives heartbeats from the server group.

   You can view the Logtail heartbeat status in the Log Service console. For more information, see View the status of a server group.

   If the heartbeat status is OK, go to the next step. If the heartbeat status is FAIL, proceed with further troubleshooting. For more information, see What can I do if no heartbeat packet is received from a Logtail client?.

2. Check whether the Logtail configuration is created.

   If the heartbeat status of Logtail is OK, check whether the Logtail configuration is created. Make sure that the path and name of monitored logs match the files that are stored on the server. The path can be a full path or a path that includes wildcards.

3. Make sure that the Logtail configuration is applied to the server group.

   Check whether the target Logtail configuration is applied to the server group. For more information, see Manage server group configurations.

4. Check collection errors.

   If Logtail is configured correctly, check whether new logs are generated in real time. Logtail only collects incremental log data. Logtail does not read log files in which no log is generated. If a log file is updated but the updates cannot be queried in Log Service, you can diagnose the problem as follows:

   - View logs of the Logtail client

The client logs include key INFO logs, all WARNING logs, and all ERROR logs. To view complete error information in real time, check the following client logs:

- Linux: */usr/local/ilogtail/ilogtail.LOG*.
- Linux: */usr/local/ilogtail/logtail_plugin.LOG*. The file contains the logs such as HTTP logs, MySQL binary logs, and MySQL query results.
- 64-bit Windows: *C:\Program Files (x86)\Alibaba\Logtail\logtail_*.log*.
- 32-bit Windows: *C:\Program Files\Alibaba\Logtail\logtail_*.log*.

- Check whether the log volume exceeds the limit.

  To collect large volumes of logs, you may need to modify the Logtail startup parameters for higher log collection throughput. For more information, see Set Logtail startup parameters.

28.1.8.1.2. What can I do if Log Service does not receive heartbeats from a Logtail client?

If Log Service does not receive heartbeats from a Logtail client, perform the steps that are described in the topic to troubleshoot the problem.

Context

After Logtail is installed on a server, the Logtail client sends heartbeats to Log Service. If the status page of the machine group shows that Log Service does not receive heartbeats from a Logtail client, it indicates that the Logtail client is not installed or disconnected from the server.

Step 1: Check whether Logtail is installed

Use the following method to check whether Logtail is installed:

- On a Linux server, run the following command:

  ```
  sudo /etc/init.d/ilogtaild status
  ```

  If the command returns `ilogtail is running`, it indicates that Logtail is installed. The following script shows an example command and response:

  ```
  [root@****************~]# sudo /etc/init.d/ilogtaild status
  ilogtail is running
  ```

- On a Windows server:
  i. Press Win+R. In the Run dialog box, enter services.msc and click **OK**.
  ii. In the **Services** window, check the status of the LogtailDaemon and LogtailWorker services. If the services are in the Running state, it indicates that Logtail is installed.

If Logtail is not installed, install it. For more information, see Install Logtail in Linux or Install Logtail in Windows. Ensure that the Log Service endpoint in the Logtail installation command corresponds to the region to which the Log Service project belongs. If Logtail is running, go to the next step.

Step 2: Check the Log Service endpoint in the Logtail installation command

When you install Logtail, you must specify a Log Service endpoint based on the region to which the Log Service project belongs. If the endpoint is incorrect or the Logtail installation command is invalid, Log Service cannot receive heartbeats from the Logtail client.

You can view the Log Service endpoint and installation method in the Logtail configuration file named *ilogtail_config.json*. The file is stored in the following path:

- Linux: */usr/local/ilogtail/ilogtail_config.json*

- 64-bit Windows: *C:\Program Files (x86)\Alibaba\Logtail\ilogtail_config.json*
- 32-bit Windows: *C:\Program Files\Alibaba\Logtail\ilogtail_config.json*

In the Logtail configuration file, check the value of the `config_server_address` parameter. This parameter specifies the Log Service endpoint. Then, check whether the Logtail client can connect to Log Service based on the endpoint. For example, if the endpoint that is recorded in the Logtail configuration file is `logtail.cn-qingdao-env25-d01.sls-pub.inter.env25.shuguang.com`, you can run the following command to check the connection:

- Linux:

```
curl logtail.cn-qingdao-env25-d01.sls-pub.inter.env25.shuguang.com
```

- Windows:

```
telnet logtail.cn-qingdao-env25-d01.sls-pub.inter.env25.shuguang.com 80
```

If the Log Service endpoint in the Logtail installation command is incorrect, re-install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows.

If the Log Service endpoint in the Logtail installation command is correct, go to the next step.

Step 3: Check the server IP addresses in the machine group

The server IP address that is obtained by a Logtail client must be configured in the machine group. Otherwise, Log Service cannot receive heartbeats or collect logs from the Logtail client. Logtail uses the following methods to obtain the IP address of a server:

- If the server is not bound with a hostname, Logtail obtains the IP address of the first network interface controller (NIC) card of the server.
- If the server is bound with a hostname, Logtail obtains the IP address that corresponds to the hostname. You can view the hostname and IP address in the */etc/hosts* file.

> **Note** You can run the **hostname** command to query the hostname.

Perform the following steps to check whether the server IP address that is obtained by the Logtail client is configured in the machine group.

1. Check the server IP address that is obtained by Logtail.

   The `ip` field in the *app_info.json* file indicates the server IP address that is obtained by Logtail. The file is stored in the following path:

   - Linux: */usr/local/ilogtail/app_info.json*
   - 64-bit Windows: *C:\Program Files (x86)\Alibaba\Logtail\app_info.json*
   - 32-bit Windows: *C:\Program Files\Alibaba\Logtail\app_info.json*

   > **Note**
   > - If the `ip` field in the `app_info.json` file is empty, Logtail cannot work. In this case, you must configure an IP address for the server and restart Logtail.
   > - The `app_info.json` file is used only to record information. If you modify the IP address in the file, the server IP address obtained by Logtail is not updated.

2. Check the server IP addresses in the machine group.

   Log on to the Log Service console. In the **Projects** section, click the project to which the machine group belongs. In the left-side navigation pane, click the **Machine Groups** icon. In the **Machine Groups** pane, click the machine group. In the **Machine Group Status** section of the **Machine Group Settings** page, check the server IP addresses.

If no server IP address in the machine group is the same as the IP address that is obtained by Logtail, perform the following step to modify the IP address configurations in the Log Service console:

- If a server IP address in the machine group is incorrect, change the IP address to the IP address that is obtained by Logtail. Then, check the heartbeat status 1 minute after you save the change.
- If you have modified the IP address of the server where Logtail is installed (for example, the */etc/hosts* file is modified), restart Logtail. After Logtail obtains the new server IP address, set a server IP address in the machine group to the value of the `ip` field in the `app_info.json` file.

You can use the following method to restart Logtail:

- On a Linux server, run the following commands:

```
sudo /etc/init.d/ilogtaild stop
sudo /etc/init.d/ilogtaild start
```

- On a Windows server:

  Press Win+R. In the Run dialog box, enter services.msc and click OK. In the **Services** window, find and restart the LogtailWorker service.

28.1.8.1.3. How do I query the local log collection statuses?

You can use Logtail to query the health status of Logtail and log collection statuses. The statuses help you troubleshoot log collection issues and customize status monitoring for log collection.

Instructions

After a Logtail client that supports the status query feature is installed, you can query the local log collection statuses by running commands on the client. For more information about how to install Logtail, see Install Logtail in Linux.

You can run the `/etc/init.d/ilogtaild -h` command on the client to check whether a client supports the feature of querying the local log collection status. If the command output includes the `logtail insight, version` keyword, it indicates that the client supports the status query feature.

```
/etc/init.d/ilogtaild -h
Usage: ./ilogtaild { start | stop (graceful, flush data and save checkpoints) | force-stop | status |
-h for help}$
logtail insight, version : 0.1.0
commond list :
       status all [index]
             get logtail running status
       status active [--logstore | --logfile] index [project] [logstore]
             list all active logstore | logfile. if use --logfile, please add project and logstore. d
efault --logstore
       status logstore [--format=line | json] index project logstore
             get logstore status with line or json style. default --format=line
       status logfile [--format=line | json] index project logstore fileFullPath
             get log file status with line or json style. default --format=line
       status history beginIndex endIndex  project logstore [fileFullPath]
             query logstore | logfile history status.
index :   from 1 to 60. in all, it means last $(index) minutes; in active/logstore/logfile/history, i
t means last $(index)*10 minutes
```

The following table describes the commands that are supported by Logtail:

Command	Function	Maximum time range that can be queried	Time window
all	Queries the status of Logtail.	Last 60 minutes	1 minute
active	Queries the active Logstores that are collecting logs and the active log files from which logs are being collected.	Last 600 minutes	10 minutes
logstore	Queries the collection status of a Logstore.	Last 600 minutes	10 minutes
logfile	Queries the collection status of a log file.	Last 600 minutes	10 minutes
history	Queries the collection status of a Logstore or log file in the query time window.	Last 600 minutes	10 minutes

Note

- The `index` parameter in the preceding commands indicates the index of the time window. Valid values: 1 to 60. The index of the latest time window is 1 and the time window ends at the current system time. If you specify a 1-minute time window, the status in the past interval of `(index, index-1]` minutes is returned. If you specify a 10-minute time window, the status in the past interval of `(10*index, 10*(index-1)]` minutes is returned.
- All commands in the preceding table is the subcommands of the status command.

Command all

Command syntax

```
/etc/init.d/ilogtaild status all [ index ]
```

Note The all command is used to query the status of Logtail. The index parameter is optional. Default value: 1.

Examples

```
/etc/init.d/ilogtaild status all 1
ok
/etc/init.d/ilogtaild status all 10
busy
```

Response

Status	Description	Priority	Troubleshooting
ok	Logtail is running as expected.	N/A	No action is required.

Status	Description	Priority	Troubleshooting
busy	The collection speed is high, and Logtail is running as expected.	N/A	No action is required.
many_log_files	A large number of log files are being collected by Logtail.	Low	You can check the Logtail configuration for log files that do not need to be collected.
process_block	The process of log parsing is blocked.	Low	You can check whether a large number of logs are generated in a short time. If you use the all command for multiple times and the returned value is always process_block, you can modify the limit of CPU usage or the limit of concurrent packet sending.
send_block	The process of sending log packets is blocked.	High	You can check whether a large number of logs are generated in a short time and the network connection is stable. If you use the all command for multiple times and the returned value is always send_block, you can modify the limit of CPU usage or the limit of concurrent packet sending.

Command active

Command syntax

```
/etc/init.d/ilogtaild status active [--logstore] index
 /etc/init.d/ilogtaild status active --logfile index project-name logstore-name
```

Note

- You can use the `active [--logstore] index` command to query all active Logstores. The `--logstore` parameter is optional.
- The command `active --logfile index project-name logstore-name` is used to query all active log files in the Logstore of a project.
- The active command is used to query log files. We recommend that you query active Logstores before querying active log files in the Logstores.

Examples

```
/etc/init.d/ilogtaild status active 1
sls-zc-test : release-test
sls-zc-test : release-test-ant-rpc-3
sls-zc-test : release-test-same-regex-3
/etc/init.d/ilogtaild status active --logfile 1 sls-zc-test release-test
/disk2/test/normal/access.log
```

Response

- If you run the `active --logstore index` command, the names of the active Logstores are returned in the following format: `project-name : logstore-name`. If you run the command `active --logfile index project-name logstore-name`, the paths of active log files are returned.

- The status of the inactive Logstores or inactive log files in the query time window is not returned.

Command logstore

Command syntax

```
/etc/init.d/ilogtaild status logstore [--format={line|json}] index project-name logstore-name
```

> **Note**
>
> - The logstore command is used to query the collection status of the specified project and Logstore in the LINE or JSON format.
> - The default value of the `--format=` parameter is `--format=line`, which indicates that the status is returned in the LINE format. Noted that the `--format=` parameter is placed after the `logstore` parameter.
> - If the Logstore specified in the preceding command does not exist or is not active in the query time window, an empty response in LINE format or the `null` value in the JSON format is returned.

Examples

```
/etc/init.d/ilogtaild status logstore 1 sls-zc-test release-test-same
time_begin_readable : 17-08-29 10:56:11
time_end_readable : 17-08-29 11:06:11
time_begin : 1503975371
time_end : 1503975971
project : sls-zc-test
logstore : release-test-same
status : ok
config : ##1.0##sls-zc-test$same
read_bytes : 65033430
parse_success_lines : 230615
parse_fail_lines : 0
last_read_time : 1503975970
read_count : 687
avg_delay_bytes : 0
max_unsend_time : 0
min_unsend_time : 0
max_send_success_time : 1503975968
send_queue_size : 0
send_network_error_count : 0
send_network_quota_count : 0
send_network_discard_count : 0
send_success_count : 302
send_block_flag : false
sender_valid_flag : true
/etc/init.d/ilogtaild status logstore --format=json 1 sls-zc-test release-test-same
{
   "avg_delay_bytes" : 0,
   "config" : "##1.0##sls-zc-test$same",
   "last_read_time" : 1503975970,
   "logstore" : "release-test-same",
   "max_send_success_time" : 1503975968,
   "max_unsend_time" : 0,
   "min_unsend_time" : 0,
   "parse_fail_lines" : 0,
   "parse_success_lines" : 230615,
   "project" : "sls-zc-test",
   "read_bytes" : 65033430,
   "read_count" : 687,
   "send_block_flag" : false,
   "send_network_discard_count" : 0,
   "send_network_error_count" : 0,
   "send_network_quota_count" : 0,
   "send_queue_size" : 0,
   "send_success_count" : 302,
   "sender_valid_flag" : true,
   "status" : "ok",
   "time_begin" : 1503975371,
   "time_begin_readable" : "17-08-29 10:56:11",
   "time_end" : 1503975971,
   "time_end_readable" : "17-08-29 11:06:11"
}
```

Response

Parameter	Description	Unit

Parameter	Description	Unit
status	The status of the Logstore.For information about Logstore statuses and actions that are required to deal with each status, see the following table.	N/A
time_begin_readable	The time when logs become readable.	N/A
time_end_readable	The time when logs become unreadable.	N/A
time_begin	The time when statistics collection starts.	Unix timestamp in seconds
time_end	The time when statistics collection ends.	Unix timestamp in seconds
project	The name of the project.	N/A
logstore	The name of the Logstore.	N/A
config	The name of the Logtail configuration, which is globally unique. The format of the name is `##1.0##` + project + `$` + config.	N/A
read_bytes	The amount of the log data that is read in the query time window.	Byte
parse_success_lines	The number of the log lines that are parsed in the query time window.	Line
parse_fail_lines	The number of the log lines that fail to be parsed in the query time window.	Line
last_read_time	The last time when logs are read in the query time window.	Unix timestamp in seconds
read_count	The number of times that the log file is read in the query time window.	Times
avg_delay_bytes	The average of difference between the actual file size and the offset generated when reading log data each time in the query time window.	Byte
max_unsend_time	The maximum period of time for which an unsent packet waits in the sending queue. An unsent packet refers to a packet that has not been sent at the end of the query time window. If no packets exist in the queue, the value is 0.	Unix timestamp in seconds

Parameter	Description	Unit
min_unsend_time	The minimum period of time for which an unsent packet waits in the sending queue. Unsent packets refer to packets that have not been sent at the end of the query time window. If no packets exist in the queue, the value is 0.	Unix timestamp in seconds
max_send_success_time	The maximum period of time when a packet waited in the sending queue.	Unix timestamp in seconds
send_queue_size	The number of the unsent packets in the sending queue at the end of the query time window.	Number of packets
send_network_error_count	The number of the packets that cannot be sent due to network errors in the query time window.	Number of packets
send_network_quota_count	The number of the packets that cannot be sent due to quota limit in the query time window.	Number of packets
send_network_discard_count	The number of the packets that are discarded due to data errors or lack of permissions.	Number of packets
send_success_count	The number of the packets that are sent in the query time window.	Number of packets
send_block_flag	Indicates whether the sending queue is blocked at the end of the query time window.	N/A
sender_valid_flag	Indicates whether the sender flag of the Logstore is valid. The value true indicates that the sender flag is valid. The value false indicates that the sender flag is invalid and disabled because of a network error or quota error.	N/A

Logstore statuses

Status	Description	Troubleshooting
ok	Logtail is running as expected.	No action is required.
process_block	The process of log parsing is blocked.	You can check whether a large number of logs are generated in a short time. If you use the all command for multiple times and the returned value is always process_block, you can Set Logtail startup parametersmodify the limit of CPU usage or of concurrent packet sending.
parse_fail	Logtail fails to parse logs.	You can check whether the format of logs is consistent with that you set in the Logtail configuration.

Status	Description	Troubleshooting
send_block	The process of sending log packets is blocked.	You can check whether a large number of logs are generated in a short time and the network connection is stable. If you use the all command for multiple times and the returned value is always send_block, you can Set Logtail startup parametersmodify the limit of CPU usage or of concurrent packet sending.

Command logfile

Command syntax

```
/etc/init.d/ilogtaild status logfile [--format={line|json}] index project-name logstore-name fileFullPath
```

Note

- The logfile command is used to query the collection status of the specified log files in the LINE or JSON format.
- The default value of the `--format=` parameter is `--format=line`, which indicates that the status is returned in the LINE format.
- If the log file specified in the command does not exist or is not active in the query time window, an empty response in the LINE format or the `null` value in the JSON format is returned.
- The `--format` parameter is placed after the `logfile` parameter.
- The value of the `filefullpath` parameter must be the full path of the log file.

Examples

```
/etc/init.d/ilogtaild status logfile 1 sls-zc-test release-test-same /disk2/test/normal/access.log
time_begin_readable : 17-08-29 11:16:11
time_end_readable : 17-08-29 11:26:11
time_begin : 1503976571
time_end : 1503977171
project : sls-zc-test
logstore : release-test-same
status : ok
config : ##1.0##sls-zc-test$same
file_path : /disk2/test/normal/access.log
file_dev : 64800
file_inode : 22544456
file_size_bytes : 17154060
file_offset_bytes : 17154060
read_bytes : 65033430
parse_success_lines : 230615
parse_fail_lines : 0
last_read_time : 1503977170
read_count : 667
avg_delay_bytes : 0
/etc/init.d/ilogtaild status logfile --format=json 1 sls-zc-test release-test-same /disk2/test/normal
/access.log
{
   "avg_delay_bytes" : 0,
   "config" : "##1.0##sls-zc-test$same",
   "file_dev" : 64800,
   "file_inode" : 22544456,
   "file_path" : "/disk2/test/normal/access.log",
   "file_size_bytes" : 17154060,
   "last_read_time" : 1503977170,
   "logstore" : "release-test-same",
   "parse_fail_lines" : 0,
   "parse_success_lines" : 230615,
   "project" : "sls-zc-test",
   "read_bytes" : 65033430,
   "read_count" : 667,
   "read_offset_bytes" : 17154060,
   "status" : "ok",
   "time_begin" : 1503976571,
   "time_begin_readable" : "17-08-29 11:16:11",
   "time_end" : 1503977171,
   "time_end_readable" : "17-08-29 11:26:11"
}
```

Response

Parameter	Description	Unit
status	The collection status of the log file in the query time window. For more information, see the status parameter in the Command logstore section.	N/A
time_begin_readable	The time when logs become readable.	N/A

Parameter	Description	Unit
time_end_readable	The time when logs become unreadable.	N/A
time_begin	The time when statistics collection starts.	Unix timestamp in seconds
time_end	The time when statistics collection ends.	Unix timestamp in seconds
project	The name of the project.	N/A
logstore	The name of the Logstore.	N/A
file_path	The path of the log file.	N/A
file_dev	The ID of the device from which the log file is collected.	N/A
file_inode	The inode of the log file.	N/A
file_size_bytes	The size of the log file that is last scanned in the query time window.	Byte
read_offset_bytes	The parsing offset of the log file.	Byte
config	The name of the Logtail configuration, which is globally unique. The format of the name is `##1.0##` + project + `$` + config.	N/A
read_bytes	The amount of the log data that is read in the query time window.	Byte
parse_success_lines	The number of the log lines that are parsed in the query time window.	Line
parse_fail_lines	The number of the log lines that fail to be parsed in the query time window.	Line
last_read_time	The last time when logs are read in the query time window.	Unix timestamp in seconds
read_count	The number of times that the log file is read in the query time window.	Times
avg_delay_bytes	The average of difference between the actual file size and the offset generated when reading log data each time in the query time window.	Byte

Command history

Command syntax

```
/etc/init.d/ilogtaild status history beginIndex endIndex project-name logstore-name [fileFullPath]
```

> **Note**
>
> - The history command is used to query the collection status of a Logstore or log file in the query time window.
> - The `beginIndex` and `endIndex` parameters specify the start and end indexes of the range of time windows that you want to query. You must ensure that `beginIndex <= endIndex`.
> - The `fileFullPath` parameter is optional. If you specify the path of a log file, the collection status of the log file is queried. If the path is not specified, the collection status of the Logstore is queried.

Examples

```
/etc/init.d/ilogtaild status history 1 3 sls-zc-test release-test-same /disk2/test/normal/access.log
        begin_time          status      read  parse_success  parse_fail      last_read_time  read_cou
nt  avg_delay    device     inode  file_size  read_offset
 17-08-29 11:26:11              ok   62.12MB         231000           0   17-08-29 11:36:11         6
71         0B     64800  22544459    18.22MB      18.22MB
 17-08-29 11:16:11              ok   62.02MB         230615           0   17-08-29 11:26:10         6
67         0B     64800  22544456    16.36MB      16.36MB
 17-08-29 11:06:11              ok   62.12MB         231000           0   17-08-29 11:16:11         6
87         0B     64800  22544452    14.46MB      14.46MB
$/etc/init.d/ilogtaild status history 2 5 sls-zc-test release-test-same
        begin_time          status      read  parse_success  parse_fail      last_read_time  read_cou
nt  avg_delay  send_queue  network_error  quota_error  discard_error  send_success  send_block  send_
valid          max_unsend          min_unsend    max_send_success
 17-08-29 11:16:11              ok   62.02MB         230615           0   17-08-29 11:26:10         6
67         0B           0             0            0              0           300       false
true   70-01-01 08:00:00   70-01-01 08:00:00   17-08-29 11:26:08
 17-08-29 11:06:11              ok   62.12MB         231000           0   17-08-29 11:16:11         6
87         0B           0             0            0              0           303       false
true   70-01-01 08:00:00   70-01-01 08:00:00   17-08-29 11:16:10
 17-08-29 10:56:11              ok   62.02MB         230615           0   17-08-29 11:06:10         6
87         0B           0             0            0              0           302       false
true   70-01-01 08:00:00   70-01-01 08:00:00   17-08-29 11:06:08
 17-08-29 10:46:11              ok   62.12MB         231000           0   17-08-29 10:56:11         6
92         0B           0             0            0              0           302       false
true   70-01-01 08:00:00   70-01-01 08:00:00   17-08-29 10:56:10
```

Response

- The collection status of the Logstore or log file in each query time window is listed in a line.
- For more information about response parameters, see the Command `logstore` and Command `logfile` sections.

Response status codes

Success code

If parameters that you specify in a command is valid (even if the queried Logstore or log file is not found), the code 0 is returned. The following section provides two examples:

```
/etc/init.d/ilogtaild status logfile --format=json 1 error-project error-logstore /no/this/file
null
echo $?
0
/etc/init.d/ilogtaild status all
ok
echo $?
0
```

Error codes

If a non-zero code is returned, it indicates that an error occurs. The following table describes the possible non-zero codes.

Code	Description	Message	Troubleshooting
10	The command is invalid or required parameters in the command are not specified.	`invalid param, use -h for help.`	You can run the `-h` command for help.
1	The value of the index parameter is not in the range from 1 to 60.	`invalid query interval`	You can run the `-h` command for help.
1	The collection status in the specified query time window cannot be queried.	`query fail, error: $(error)` . For more information, visit errno.	The startup time of Logtail is earlier than the query time window. Otherwise, you can submit a ticket for help.
1	The start time of querying is out of the query time window.	`no match time interval, please check logtail status`	You can check whether Logtail is running. If yes, you can submit a ticket for help.
1	No logs exist in the specified query time window.	`invalid profile, maybe logtail restart`	You can check whether Logtail is running. If yes, you can submit a ticket for help.

Examples

```
/etc/init.d/ilogtaild status nothiscmd
invalid param, use -h for help.
echo $?
10
/etc/init.d/ilogtaild status/all 99
invalid query interval
echo $?
1
```

Scenarios

You can query the overall status of Logtail, and specific metrics by querying the collection status during collection. You can customize a mechanism to monitor the log collection status based on the queried information.

Monitor the status of Logtail

You can monitor the status of Logtail by using the `all` command.

For example, you can run the command every minute to query Logtail status. If the `process_block`, `send_block` or `send_error` value is returned for 5 consecutive minutes, an alert is triggered.

You can adjust the alert duration and monitoring scope based on the priorities of the collected log files.

Monitor the log collection status

You can monitor the log collection status of a Logstore by using the `logstore` command.

For example, you can run the `logstore` command every 10 minutes to query the collection status of the Logstore. If the value of the `avg_delay_bytes` parameter exceeds 1 MB (1024 × 1024 bytes) or the value of the `status` parameter is not `ok`, an alert is triggered.

You can adjust the alert threshold for the `avg_delay_bytes` metric based on the size of data that is generated during the log collection.

Check whether Logtail has finished collecting log files

You can check whether Logtail has finished collecting log files by using the `logfile` command.

After Logtail stops collecting log files, you can run the `logfile` command every 10 minutes to query the status of the log file. If the value of the `read_offset_bytes` parameter is the same as that of the `file_size_bytes` parameter, it indicates that the log file is collected.

Troubleshoot log collection issues

If log collection latency occurs on a server, you can use the `history` command to query the status history of log collection.

1. The value of the `send_block_flag` parameter is true.This indicates that the log collection is blocked because of unstable network connections.
   - If the value of the `send_network_quota_count` parameter is greater than 0, split shards in the Logstore. For more information, see Split a shard.
   - If the value of the `send_network_error_count` parameter is greater than 0, check the network connections.
   - If no network error occurs, adjust the limit of concurrent packet sending and data transfer speed of Logtail.
2. The parameters related to packet sending are set to appropriate values. However, the value of the `avg_delay_bytes` parameter is large.
   - Use the value of the `read_bytes` parameter to calculate the average speed of parsing logs, and then determine whether a large amount of data is transferred during log collection based on the average speed.
   - Adjust the resource usage limits for Logtail.
3. The value of the `parse_fail_lines` parameter is greater than 0.

   Check whether the regular expression can match all required log fields as expected.

28.1.8.1.4. How do I test a regular expression?

If you select the full regex mode when you configure Logtail to collect and parse text logs, you must specify a regular expression based on your sample log entries. This topic describes how to test a regular expression.

Context

To test a regular expression that you have specified in the Log Service console, you can click **Validate** in the console and check the results as follows:

- For the regular expression that matches the first line of logs, check whether the regular expression can match the expected number of log entries.
- For the fields extracted by the regular expression, check whether the value of each field meets your expectations.

If you want to validate more items and test a regular expression, you can use online tools such as regex101.com and regextester.com. You can copy and paste the regular expression that is generated by Log Service to an online tool, and specify a sample log entry as the test string.

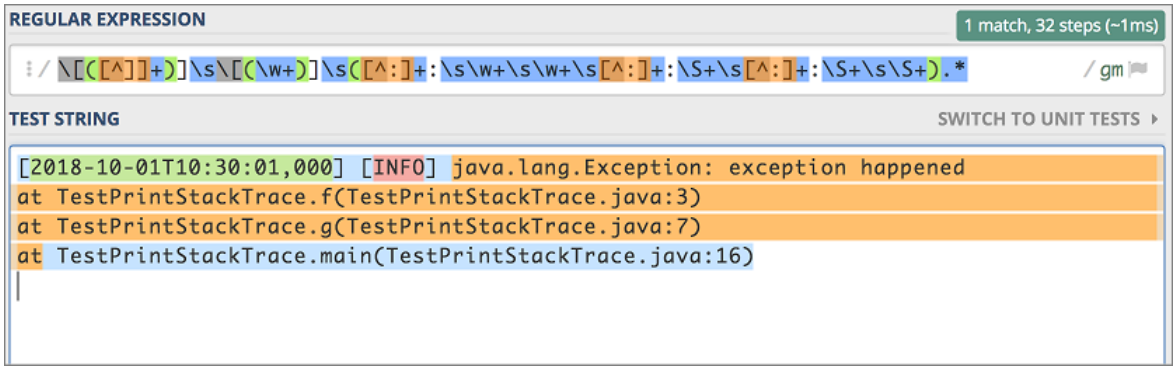
If you use the full regex mode, Log Service automatically generates a regular expression based on a sample log entry. However, the regular expression may fail to match the message field in multi-line log entries as expected. The following example describes how to use the regex101.com tool to test the regular expression.

Procedure

1. Copy the generated regular expression.
2. Visit the regex101.com website.
3. Paste the regular expression in the **REGULAR EXPRESSION** field.

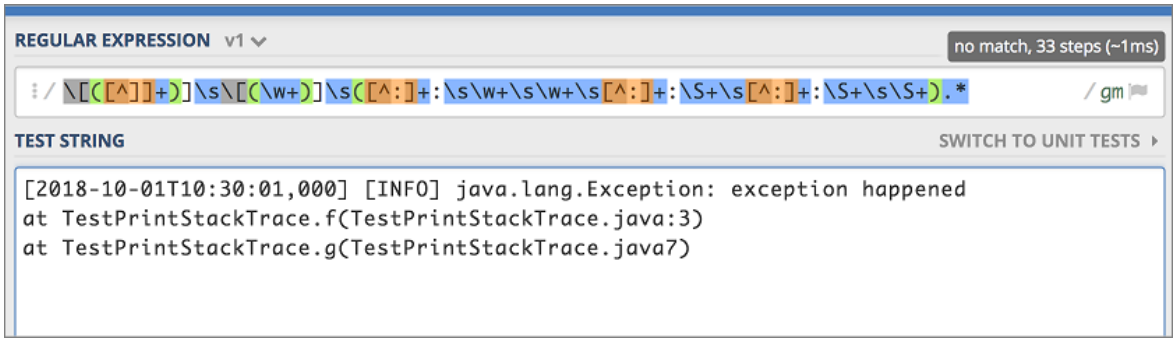
   On the right side of the page, you can view the explanation of the regular expression.
4. In the **TEST STRING** field, paste a log sample entry.

   In the following figure, the log contents that are included in the message field are highlighted in orange, and the log contents that are not included are highlighted in blue. The figure shows that the substring following the `at` word is not included in the `message` field. Therefore, this regular expression does not match fields in the sample log entry as expected and cannot be used to collect log data.



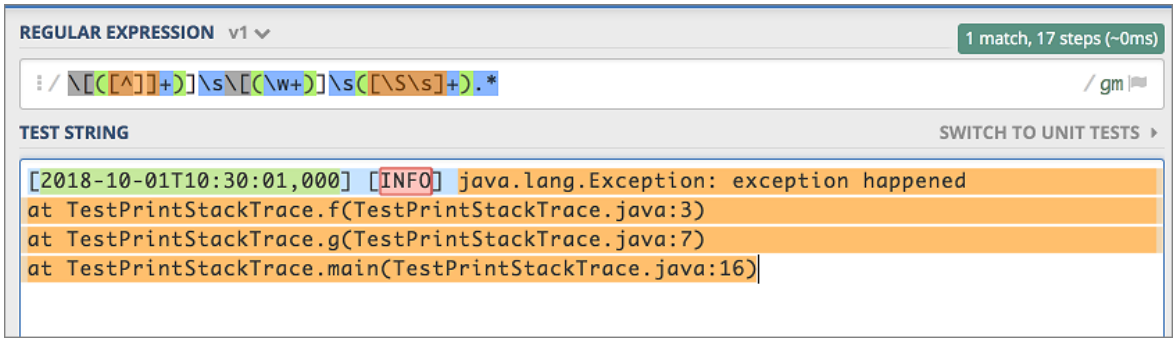
5. Check whether the regular expression can match fields in a sample log entry with two colons as expected.

   The following figure shows that the regular expression fails to match fields in the sample log entry as expected.

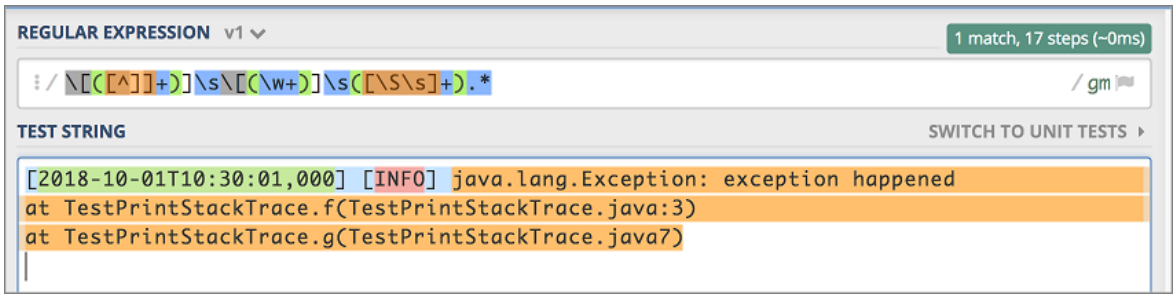


6. Replace the last subexpression in the regular expression with `[\S\s]+`, and check whether the regular expression can match fields in the sample log entries as expected.

   The following figure shows how the modified regular expression matches the substring following the `at` word.



The following figure shows how the modified regular expression matches the sample log entry with two colons.



You can follow the preceding instructions to test your regular expression. After you validate the regular expression, you can apply it to a Logtail configuration.

28.1.8.1.5. How do I optimize regular expressions?

You can optimize regular expressions to improve the Logtail performance.

When you optimize regular expressions, we recommend that you follow these rules:

- Use precise characters

  We recommend that you do not use the wildcard characters `.*` in a regular expression to match fields in log entries. Using wildcard characters may lead to mismatches and low matching performance. For example, if a field that you want to match only consists of letters, use `[A-Za-z]`.

- Use appropriate quantifiers

  We recommend that you do not use plus signs (+) or asterisks (*). For example, you can use `\d` instead of `\d+` or `\d{1,3}` to match the IP address.

- Test and modify regular expressions

  You can visit the **regex101.com** website to test and modify a regular expression to decrease the time required to match log entries.

28.1.8.1.6. How do I use the full regex mode to collect log entries in multiple formats?

The full regex mode requires that log entries to be collected be in the same format. Therefore, if you want to collect log entries that are in multiple formats, you must use the schema-on-write or schema-on-read solution.

Taking Java logs as an example, the following section lists the types of error log entry and normal log entry.

- Multi-line WARNING log entries
- Simple text INFO log entries
- Key-value DEBUG log entries

```
[2018-10-01T10:30:31,000] [WARNING] java.lang.Exception: another exception happened
    at TestPrintStackTrace.f(TestPrintStackTrace.java:3)
    at TestPrintStackTrace.g(TestPrintStackTrace.java:7)
    at TestPrintStackTrace.main(TestPrintStackTrace.java:16)
[2018-10-01T10:30:32,000] [INFO] info something
[2018-10-01T10:30:33,000] [DEBUG] key:value key2:value2
```

To collect log entries of these types, you can use the following solutions:

- Schema-on-write: To extract log fields, you must apply multiple Logtail configurations with different regular expressions to a log file.

  > **Note** However, Logtail cannot apply multiple Logtail configurations directly to the same log file. Therefore, you must set up multiple symbolic links for the directory in which the log file resides. Each Logtail configuration applies to a symbolic link to collect log entries in a specific format.

- Schema-on-read: you can use a common regular expression to collect log entries in different formats.

  For example, if you want to collect log entries in multiple formats, you can configure a regular expression that matches the time and log level fields as the first line, and specify the rest of the log entries as the log message. If you want to parse the message, create an index for the message, specify a regular expression to extract log messages, and then extract target fields.

  > **Note** We recommend that you use this solution only for scenarios in which tens of millions of log entries are collected, or fewer.

28.1.8.1.7. How do I set the time format for logs?

You must be familiar with the following rules before setting the time format for logs in Logtail configurations.

- The unit of the timestamp in Log Service is seconds. Therefore, you cannot set the unit as milliseconds or microseconds.
- You only need to set the time field. Other parameters are not required.

The following section lists commonly used formats:

```
Custom1 2017-12-11 15:05:07
%Y-%m-%d %H:%M:%S
Custom2 [2017-12-11 15:05:07.012]
[%Y-%m-%d %H:%M:%S
RFC822     02 Jan 06 15:04 MST
%d %b %y %H:%M
RFC822Z    02 Jan 06 15:04 -0700
%d %b %y %H:%M
RFC850      Monday, 02-Jan-06 15:04:05 MST
%A, %d-%b-%y %H:%M:%S
RFC1123     Mon, 02 Jan 2006 15:04:05 MST
%A, %d-%b-%y %H:%M:%S
RFC3339     2006-01-02T15:04:05Z07:00
%Y-%m-%dT%H:%M:%S
RFC3339Nano 2006-01-02T15:04:05.999999999Z07:00
%Y-%m-%dT%H:%M:%S
```

28.1.8.1.8. How do I configure non-printable characters in a sample log?

This topic describes how to configure non-printable characters in a sample log entry.

Context

Log Service allows you to specify a non-printable character as the delimiter or quote to collect logs. Non-printable characters are those whose decimal ASCII codes are in the range of 1 to 31 and 127. If you want to specify a non-printable character as the delimiter or quote, you must find the hexadecimal ASCII code of this character and enter this character in the following format: `0x` *the hexadecimal ASCII code of the non-printable character*. For example, a sample log entry is `123456780`. You can set 0x01 as the delimiter and 0x02 as the quote, and then enter a non-printable character 0x01 between the digits 5 and 6.

Procedure

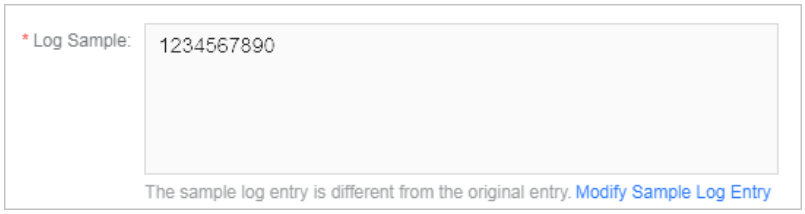
1. Log on to the Log Service console.
2. Right-click the blank space on the browser and select **Inspect** from the shortcut menu.
3. Click the **Console** tab on the page that appears.
4. Enter `"\x01"` in the code editor and press Enter.
5. Copy the returned result.

   A non-printable character is enclosed in quotation marks (").

6. Paste the returned result between the digits 5 and 6.

7. Delete the quotation marks between the digits 5 and 6.

   Then, a non-printable character is configured in a sample log entry.



28.1.8.1.9. How do I troubleshoot errors during container log collection?

Perform the steps that are described in this topic to troubleshoot an error that occurs when you use Logtail to collect logs from common containers or containers in a Kubernetes cluster.

Related O&M operations

- Log on to a Logtail container
- View the operational logs of Logtail
- Ignore the stdout logs of a Logtail container
- View the status of Log Service components in a Kubernetes cluster
- View the version number, IP address, and startup time of Logtail

Troubleshoot an error if Log Service does not receive heartbeats from Logtail clients

Perform the following steps to check whether Logtail is installed:

1. In the machine group, count the number of the servers whose heartbeat status is OK.
   i. Log on to the Log Service console.
   ii. Click the project to which the machine group belongs.
   iii. In the left-side navigation pane, click **Machine Groups**.
   iv. In the **Machine Groups** pane, click the name of the machine group.
   In the **Machine Group Status** section, count the number of the servers whose heartbeat status is **OK**.
2. Count the number of worker nodes in the cluster.
   Run the `kubectl get node | grep -v master` command to query the work nodes in the cluster. Count the number of the work nodes that are returned.

```
$kubectl get node | grep -v master
NAME                                STATUS    ROLES     AGE       VERSION
cn-hangzhou.i-bp17enxc2us3624wexh2   Ready     <none>    238d      v1.10.4
cn-hangzhou.i-bp1ad2b02jtqd1shi2ut   Ready     <none>    220d      v1.10.4
```

3. Check whether the number of servers whose heartbeat status is **OK** in the machine group is equal to the number of worker nodes in the cluster. Troubleshoot the error based on the check result.
   - The number of servers whose heartbeat status is OK is equal to the number of worker nodes. This means that the heartbeat status of all of the servers in the machine group is **Failed**.
     - If Logtail is installed into a common container, check whether the values of the *${your_region_name}*, *${your_aliyun_user_id}*, and *${your_machine_group_user_defined_id}* parameters are correct. For information about how to set these parameters, see Collect standard Docker logs.
     - If Logtail is installed into a container in a Container Service for Kubernetes cluster, submit a ticket.
     - If Logtail is installed into a container in a user-created Kubernetes cluster, check whether the values of the *{your-project-suffix}*, *{regionId}*, *{aliuid}*, *{access-key-id}*, and *{access-key-secret}* parameters are correct. If the value of a parameter is incorrect, run the `helm del --purge alibaba-log-controller` command to delete the installation package and re-install Logtail. For information about how to set these parameters, see Collect Kubernetes logs.
   - The number of servers whose heartbeat status is OK is less than the number of worker nodes.

a. Check whether you used a YAML file to manually deploy a DaemonSet.

   Run the `kubectl get po -n kube-system -l k8s-app=logtail` command to perform the check. If the command returns pod information, it indicates that you manually deployed a DaemonSet by using a YAML file.

b. Download the latest version of the Logtail DaemonSet template.

c. Set the *${your_region_name}*, *${your_aliyun_user_id}*, and *${your_machine_group_name}* parameters to the values that are specific to your environment.

d. Run the `kubectl apply -f ./logtail-daemonset.yaml` command to update the DaemonSet YAML file.

Submit a ticket if the error persists.

Troubleshoot an error if Log Service collects no logs from containers

If no log is displayed in the **Consumption Preview** pane or on the **Search & Analysis** page of a Logstore, it indicates that Log Service does not collect logs from the machine group of the Logstore. Check the status of the containers that correspond to the servers in the machine group. If the containers are working as expected, perform the following steps to troubleshoot the error:

1. Check the heartbeat status of the servers in the machine group.
2. Check whether the parameter settings in the Logtail configuration files are correct.

   Check whether the values of the IncludeLabel, ExcludeLabel, IncludeEnv, and ExcludeEnv parameters in the Logtail collection configuration files meet your requirements.

   > **Note** The `IncludeLabel` or `ExcludeLabel` parameter specifies whether to include or exclude the container images to which specified labels are attached. You can retrieve a list of container image labels by running the docker inspect command. The labels are not the labels that are defined by using Kubernetes. To check whether the parameter settings are correct in a Logtail configuration file, delete the IncludeLabel, ExcludeLabel, IncludeEnv, and ExcludeEnv parameters from the file. If Log Service can collect logs from the containers after the parameters are deleted, it indicates that the settings of the parameters are incorrect.

3. Check other items.

   Log Service does not collect logs from containers in the following scenarios:

   - Log files are not updated.
   - The log files of a container are stored in locations that are neither the default storage nor a storage attached to the container.

Log on to a Logtail container

- Common container
  i. Run the `docker ps | grep logtail` command on the host to search for the Logtail container.
  ii. Run the `docker exec -it ****** bash` command to log on to the container.

```
$docker ps | grep logtail
223fbd3ed2a6e        registry.cn-hangzhou.aliyuncs.com/log-service/logtail
"/usr/local/ilogta..."   8 days ago          Up 8 days                                  logtail-iba
$docker exec -it 223fbd3ed2a6e bash
```

- Container in a Kubernetes cluster
  i. Run the `kubectl get po -n kube-system | grep logtail` command to search for the pod where the Logtail container resides.
  ii. Run the `kubectl exec -it -n kube-system ****** bash` command to log on to the pod.

```
$kubectl get po -n kube-system | grep logtail
logtail-ds-g5wgd                                        1/1       Running    0          8d
logtail-ds-slpn8                                        1/1       Running    0          8d
$kubectl exec -it -n kube-system logtail-ds-g5wgd bash
```

View the operational logs of Logtail

The operational logs of Logtail are saved in the files named *ilogtail.LOG* and *logtail_plugin.LOG* under the */usr/local/ilogtail/* directory of a Logtail container.

1. Log on to a Logtail container.
2. Open the */usr/local/ilogtail/* directory.

   ```
   cd /usr/local/ilogtail
   ```

3. View the *ilogtail.LOG* and *logtail_plugin.LOG* files.

   ```
   cat ilogtail.LOG
   cat logtail_plugin.LOG
   ```

Ignore the stdout logs of a Logtail container

The standard output (stdout) logs of a Logtail container are useless for troubleshooting. Ignore the following stdout logs:

```
start umount useless mount points, /shm$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fbe2bdb95d13b1e110172ef57fe
840c82155/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44be
ab6e69718/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640b1
e16c22dbe/merged: must be superuser to unmount
......
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
ilogtail is running
```

View the status of Log Service components in a Kubernetes cluster

Run the `helm status alibaba-log-controller` command to view the status of Log Service components in a Kubernetes cluster.

View the version number, IP address, and startup time of Logtail

View the information in the *app_info.json* file under the */usr/local/ilogtail/* directory of the Logtail container. For example, you can run the following command to view the content of the file:

```
kubectl exec logtail-ds-gb92k -n kube-system cat /usr/local/ilogtail/app_info.json
{
   "UUID" : "",
   "hostname" : "logtail-gb92k",
   "instance_id" : "0EBB2B0E-0A3B-11E8-B0CE-0A58AC140402_10.10.10.10_1517810940",
   "ip" : "10.10.10.10",
   "logtail_version" : "0.16.2",
   "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
   "update_time" : "2018-02-05 06:09:01"
}
```

28.1.8.2. Log search and analysis

28.1.8.2.1. FAQ about log query

This topic describes the common issues that may occur when you query log data in the Log Service console. It also includes solutions to these issues.

How do I identify the source server from which Logtail collects logs during a query?

If a server group uses IP addresses as its identifier when logs are collected by using Logtail, servers in the server group are distinguished from one another by internal IP addresses. When querying logs, you can use the hostname and custom IP address to identify the source server from which logs are collected.

For example, you can use the following statement to count the times different hostnames appear in logs:

> **Note** You must enable the index feature for the Logstore and enable the statistics feature for the __tag__:__hostname__ field in advance.

```
* | select "__tag__:__hostname__" , count(1) as count group by "__tag__:__hostname__"
```

How do I query IP addresses in logs?

You can use the exact match method to query IP addresses in logs. For example, you can specify IP addresses to query log data that includes or excludes the specified IP addresses. However, you cannot use the partial match method to query log data related to specified IP addresses. This is because decimal points contained in an IP address are not default delimiters in Log Service. If you want to use the partial match method, you can configure the decimal point as a delimiter for indexes. For example, you can use the SDK to download data and then use regular expressions or the string.indexof method in the code.

For example, you use the following statement to query projects that meet the specified conditions.

```
not ip:121.42.0 not status:200 not 360jk not DNSPod-Monitor not status:302 not jiankongbao
        not 301 and status:403
```

The retrieved log data still contains 121.42.0.x. An IP address such as 121.42.0.x is taken as a word in Log Service. To include or filter out 121.42.0.x in the query result, you must specify 121.42.0.x in the query statement. If you specify 121.42.0 in the query statement, you cannot retrieve log data that includes or excludes the keyword.

How do I query log data by using a keyword that contains a whitespace character?

If you use a keyword that contains a whitespace character to query log data, log data that contains the part of the keyword on the left or right of the whitespace character is retrieved. You can enclose the keyword that contains a whitespace character in double quotation marks (""). Then the entire enclosed content is regarded as a keyword to query the log data that you expect.

For example, you want to query log data that contains the keyword `POS version` from the following log data:

```
post():351]: device_id: BTAddr : B6:xF:xx:65:xx:A1 IMEI : 35847xx2
2xx81x9 WifiAddr : 4c:xx:0e:xx:4e:xx | user_id: bb07263xxd2axx43xx9exxe
a26e39e5f POS version:903
```

If you use `POS version` as the keyword, log data that contains `POS` and `version` is retrieved. This result does not meet your expectations. If you use `"POS version"` as the keyword, log data that contains the keyword `POS version` is retrieved.

How do I use two query conditions to query log data?

You can enter two query conditions at one time to query log data that you want.

For example, you want to query log data whose status field value is neither OK nor Unknown in a Logstore. You can use the `not OK not Unknown` statement to retrieve the expected result.

How can I query collected logs in Log Service?

You can use one of the following methods to query logs in Log Service:

- Use the Log Service console.
- Use the SDK.
- Use the Restful API.

28.1.8.2.2. What can I do if no log data is retrieved?

When you use the log search and analytics feature of Log Service to query data, you may not retrieve the data you want. In this case, you can troubleshoot the problem as follows:

Log collection failure

If log data fails to be collected by Log Service, the target log data cannot be queried. Check whether log data is available on the consumption preview page of the target Logstore.

If data is available, log data is collected by Log Service.

If data is unavailable, possible causes are as follows:

- The log source does not generate log data.

  In this case, no logs can be sent to Log Service. Check your log source.

- Logtail has no heartbeat.

  On the **Server Group Settings** page, check whether the relevant server has a heartbeat in the Server Group Status section. If it has no heartbeat, troubleshoot the problem. For more information, see What can I do if no heartbeat packet is received from a Logtail client?

- The monitoring file is not written in real time.

  If the monitoring file is written in real time, you can open the */usr/local/ilogtail/ilogtail.LOG* file to view the error message. Common error messages are as follows:

  - parse delimiter log fail: The error message is returned because an error has occurred when Log Service collects logs in the delimiter mode.
  - parse regex log fail: The error message is returned because an error has occurred when Log Service collects logs in the regex mode.

Delimiter setting errors

View the configured delimiters. Check whether you can use a keyword to query log data after the log content is split by using a delimiter. For example, the default delimiters `, ; = ( ) [ ] { } ? @ & < > / : '` are used. If a log entry contains abc"defg,hij, it is split into abc"defg and hij. In this case, you cannot retrieve this log entry by searching for abc.

Fuzzy match is also supported. For more information, see Query syntax.

> **Note**
>
> - To save your indexing cost, Log Service has optimized the index feature. If you configure an index for a field, full-text indexing is ineffective for the key of this field. For example, an index is configured for a log field whose key is message, and a whitespace character is used as a delimiter. To use a whitespace character as a delimiter, put it in the middle of delimiters that you have configured for an index. You can retrieve the log entries that contain "message: this is a test message" by searching for the key-value-pair-formatted keyword message:this. However, if you use the keyword this to query the log entries, you cannot retrieve the data because an index is configured for the message field and full-text indexing is ineffective.
> - You can create indexes or modify existing indexes. However, new or modified indexes take effect only for new data.
>
>   You can click Index Attributes to check whether the configured delimiters meet the requirements.

Other reasons

If log data is available, modify the time range of the query and try again. Log Service allows you to preview log data in real time. Due to a maximum latency of one minute, we recommend that you query log data at least one minute after logs are generated.

28.1.8.2.3. What are the differences between log consumption and log search and analytics?

Both the log consumption and log search and analytics features provided by Log Service need to read log data. The log consumption feature provides log collection and distribution channels. In contrast, the log search and analytics feature allows you to query log data.

Both the log consumption and log search and analytics features need to read log data:

Log collection and consumption (LogHub): provides public channels for log collection and distribution. It reads and writes full data in first-in, first-out (FIFO) order, which is similar to Kafka.

- Each Logstore has one or more shards. Data is written to a random shard.
- You can read multiple log entries at a time from a specified shard based on the order in which the log entries were written to the shard.
- You can set the start position (cursor) to pull log entries from shards based on the time when Log Service receives the log entries.

Log search and analytics: enables you to set conditions to search and analyze large amounts of log data based on LogHub.

- This feature allows you to search for required data based on query conditions.
- This feature allows you to include a combination of Boolean keywords AND, NOT, and OR and SQL statements in a query.
- This feature is independent of shards.

The following table lists the differences between the log search and analytics feature and the LogHub feature.

Feature	Log search and analytics (LogSearch)	LogHub
Search by keyword	Supported.	Not supported.
Data read (a small amount of data)	Fast.	Fast.
Data read (full data)	Slow. LogSearch reads 100 log entries in 100 milliseconds. This method is not recommended.	Fast. LogHub reads 1 MB of log data in 10 milliseconds. This method is recommended.
Data read by topic	Yes.	No. Data is identified only by shard.
Data read by shard	No. Data in all shards of a Logstore is queried.	Yes. You need to specify a shard each time to read data.
Price	Relatively high.	Low.
Scenario	Monitoring, problem investigation, and analysis.	Full data processing scenarios, such as stream computing and batch processing.

28.1.8.2.4. How do I resolve common errors returned in log data queries?

Common errors returned in log data queries are as follows:

line 1:44: Column 'my_key_field' cannot be resolved;please add the column in the index attribute

- Cause

  The `my_key_field` key cannot be included in the query statement because it does not exist.

- Solution

  In the upper-right corner of the Search & Analysis page, click Index Attributes to create an index for this field and enable the statistics feature for this field.

Column 'xxxxline' not in GROUP BY clause;please add the column in the index attribute

- Cause

  You use the GROUP BY clause and include a non-aggregated field in a SELECT statement. However, this field is not specified in the GROUP BY clause. For example, the key1 field in the `select key1, avg(latency) group by key2` statement is not specified in the GROUP BY clause.

- Solution

  An example correct statement is `select key1,avg(latency) group by key1,key2`.

sql query must follow search query,please read syntax doc

- Cause

  You do not include a filtering condition in a query statement, for example, `select ip,count(*) group by ip`.

- Solution

  An example correct statement is `*|select ip,count(*) group by ip`.

please read syntax document,and make sure all related fields are indexed. error after select .error detail:line 1:10: identifiers must not start with a digit; surround the identifier with double quotes

- Cause

  The column name or variable name referenced in an SQL statement starts with a number and does not comply with the rules.

- Solution

  Change the name so the name starts with a letter.

please read syntax document,and make sure all related fields are indexed. error after select .error detail:line 1:9: extraneous input '' expecting

- Cause

  Misspelled words exist in the query statement.

- Solution

  Correct the misspelled words.

key (category) is not config as key value config,if symbol : is in your log,please wrap : with quotation mark "

- Cause

  No index is configured for the category field. It cannot be used in a query statement.

- Solution

  Configure an index for this field in the index attributes. For more information, see Enable the index feature and configure indexes for a Logstore.

Query exceeded max memory size of 3GB

- Cause

  The size of memory used by the current query exceeds 3 GB. The common cause is that a large number of values are still returned in the query result after you use the GROUP BY clause to remove duplicates.

- Solution

  Reduce the number of keys specified in the GROUP BY clause.

ErrorType:ColumnNotExists.ErrorPosition,line:0,column:1.ErrorMessage:line 1:123: Column '__raw_log__' cannot be resolved; it seems __raw_log__ is wrapper by ";if __raw_log__ is a string ,not a key field, please use '__raw_log__'

- Cause

  The `my_key_field` key cannot be included in the query statement because it does not exist.

- Solution

  In the upper-right corner of the Search & Analysis page, click Index Attributes to create an index for this field and enable the statistics feature for this field.

28.1.8.2.5. Why data queries are inaccurate?

This topic describes the causes for inaccurate data queries. It also includes solutions to these issues.

When you search and analyze log data, the message **The results are inaccurate** may prompt in the console. This indicates that the returned result is inaccurate because some log data in a Logstore was not queried.

Possible causes include:

The time range for queries is excessive.

- Cause

  The time range for a query is excessively wide, for example, three months or a year. In this case, Log Service cannot scan all log data generated within this time period for one query.

- Solution

  Narrow down the query time range and perform multiple queries.

Query statements are complex.

- Cause

  The query statement is exceedingly complex or contains multiple frequently used words. In this case, Log Service cannot scan all related log data or read the query results at one time.

- Solution

  Narrow down the query scope and perform multiple queries.

The SQL computing needs to read an excessively large amount of data.

- Cause

  The SQL computing needs to read an excessively large amount of data. In this case, query results are likely to become inaccurate. A maximum of 1 GB of data can be read from each shard. For example, if the SQL computing needs to read strings from multiple columns, which exceed the threshold data volume, inaccurate query results will be returned.

- Solution

  Narrow down the query scope and perform multiple queries.

28.1.8.2.6. How do I configure indexes for historical log data?

You cannot directly configure indexes for historical log data in Log Service. To configure indexes for historical log data, you can use DataWorks or the command line interface (CLI) to move data into another Logstore.

Indexes take effect on log data that is collected after the indexes are configured. You cannot use indexes to search and analyze historical log data. To configure indexes for historical log data, you can use either of the following two methods:

- Configure indexes in a new Logstore and then use DataWorks to move data into the Logstore.

  After you configure indexes in a new Logstore, you can use DataWorks to move historical log data from the Logstore where it is stored to the new Logstore. Then you can use the configured indexes to search and analyze the data.

- After you configure indexes in a new Logstore, you can use the CLI to export historical log data from the Logstore where it is stored to the new Logstore.

> **Note** The preceding two methods copy historical log data and then export the data into a Logstore. They do not change or delete the data.

28.1.8.3. Alarm

28.1.8.3.1. FAQ about alerts

This topic describes the common issues that may occur when you configure alerts in the Log Service console. It also includes solutions to these issues.

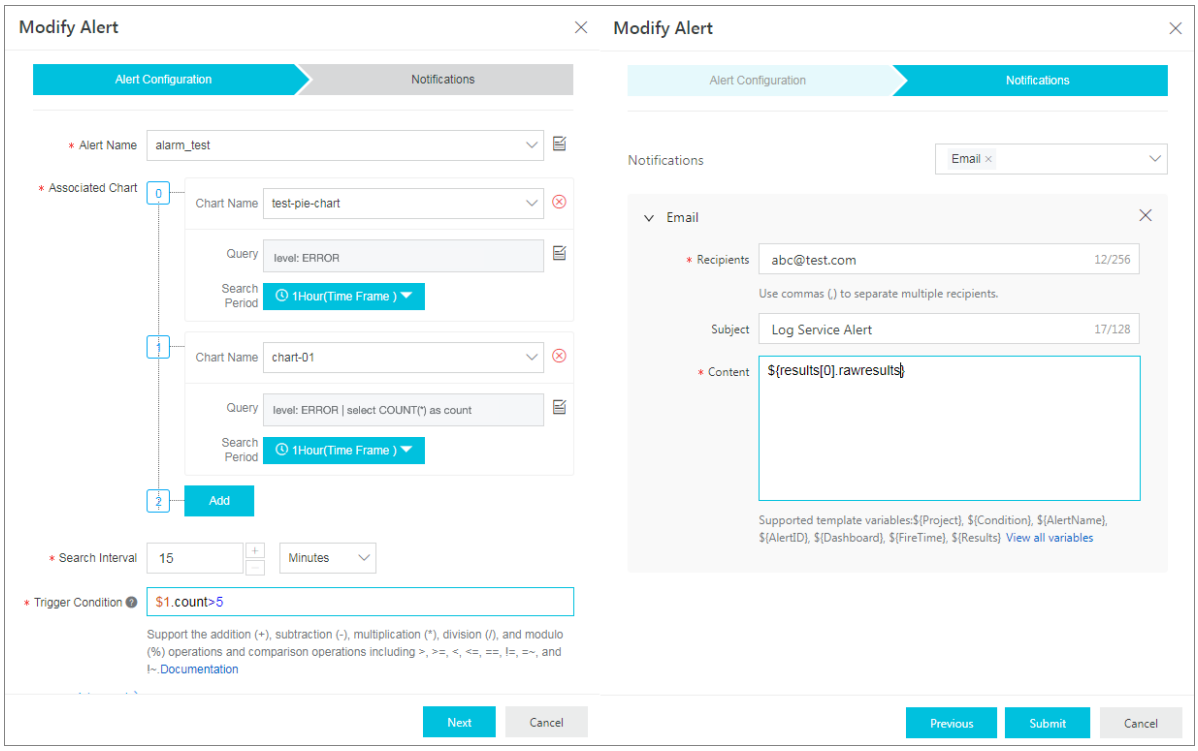
How can I include the raw error log entries in the notification content?

- Issue

  More than five error log entries were generated in the past five minutes, which triggered an alert. How can I include the raw error log entries in notifications that were sent when the alert was triggered?

- Solution
  - Associated query statements
    - Number 0: `level:ERROR`
    - Number 1: `level:ERROR | select COUNT(*) as count`
  - Trigger condition: `$1.count > 5`
  - Notification content: `${results[0].rawresults}`

- Configuration examples



29.Apsara Stack DNS

29.1. User Guide

29.1.1. What is Apsara Stack DNS?

Apsara Stack DNS is a service that runs on Apsara Stack to resolve domain names. You can configure rules to map domain names to IP addresses. Apsara Stack DNS then distributes domain name requests from clients to cloud resources, business systems on your internal networks, or the business resources of Internet service providers (ISPs).

Apsara Stack DNS provides DNS resolution in VPCs. You can perform the following operations on your VPC by using Apsara Stack DNS:

- Manage internal domain names.
- Manage DNS records of internal domain names.
- Manage forwarding configurations.
- Manage recursive resolution configurations.

29.1.2. User roles and permissions

Role	Permission
System administrator	A user of this role has read, write, and execute permissions on all level-1 organization resources, global resources, and system configurations.
Level-1 organization administrator	A user of this role has read, write, and execute permissions on level-1 organization resources to which the user belongs, but does not have permissions on level-1 organization resources, global resources, or system configurations to which other users belong.
Lower-level organization administrator	A user of this role does not have permissions on Apsara Stack DNS. The user does not have permissions on level-1 organization resources to which the user belongs, and does not have permissions on level-1 organization resources, global resources, or system configurations to which other users belong.
Resource user	A user of this role does not have permissions on Apsara Stack DNS. The user does not have permissions on level-1 organization resources to which the user belongs, and does not have permissions on level-1 organization resources, global resources, or system configurations to which other users belong.
Other roles	A user of this role does not have permissions on Apsara Stack DNS. The user does not have permissions on level-1 organization resources to which the user belongs, and does not have permissions on level-1 organization resources, global resources, or system configurations to which other users belong.

29.1.3. Log on to the Apsara Stack DNS console

This topic describes how to log on to the Apsara Stack DNS console by using Google Chrome.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, choose **Products > Networking > Apsara Stack DNS**.

29.1.4. Internal DNS resolution management

Internal DNS resolution management allows you to manage global internal domain names, global forwarding configurations, and global recursive resolution configurations that you have created in Apsara Stack.

29.1.4.1. Global internal domain names

29.1.4.1.1. Overview

All the operations of this feature require administrator privileges.

29.1.4.1.2. View an internal domain name

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domains > Global Internal Domains**.
3. In the **Domain Name** search box, enter the domain name that you want to view.
4. Click **Search**.

   The search result is displayed.

29.1.4.1.3. Add a domain name

This topic describes how to add a domain name in the Apsara Uni-manager Management Console.

Procedure

1. Log on to the Apsara Stack DNS console.

2. In the left-side navigation pane, choose **Internal Domain Names > Internal Domain**.
3. Click **Add Domain Name**.
4. In the dialog box that appears, set **Internal Domain**.
5. Click **OK**.

29.1.4.1.4. Add a description for a domain name

This topic describes how to add a description for a domain name in the Apsara Uni-manager Management Console.

Context

You can add a description for a domain name to help you easily identify it. For example, you can add a hostname or internal system information to describe a domain name.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domain Names > Internal Domain**.
3. Find the domain name for which you want to add a description, click the  icon in the Actions column, and then select **Description**.
4. In the dialog box that appears, enter a description.
5. Click **OK**.

29.1.4.1.5. Delete a domain name

This topic describes how to delete a domain name in the Apsara Uni-manager Management Console.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domain Names > Internal Domain**.
3. Find the domain name that you want to delete, click the  icon in the **Actions** column, and then select **Delete**.
4. In the message that appears, click **OK**.

29.1.4.1.6. Delete multiple domain names

This topic describes how to delete unnecessary domain names at a time in the Apsara Uni-manager Management Console.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domain Names > Internal Domain**.
3. Select one or more domain names that you want to delete and click **Delete** in the upper-right corner.
4. In the message that appears, click **OK**.

29.1.4.1.7. Configure DNS records

This topic describes how to configure DNS records in the Apsara Uni-manager Management Console.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domain Names > Internal Domain**.
3. Find the domain name for which you want to configure DNS records, click the icon in the Actions column, and then select **Configure DNS Records**.
4. In the upper-right corner of the **Configure DNS Records** page, click **Add DNS Record**.
5.

29.1.4.1.8. View a resolution policy

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domains > Global Internal Domains**.
3. Find the target domain name, click the icon in the Actions column, and then select **Configure DNS Records**.
4. On the page that appears, select the domain name for which you want to configure DNS records, and click **Weight** in the **Resolution Policy** column.
5. On the page that appears, view the details of **Resolution Policy**.

29.1.4.2. Global forwarding configurations

29.1.4.2.1. Global forwarding domain names

29.1.4.2.1.1. Overview

All operations of this feature require administrator privileges.

Apsara Stack DNS forwards specific domain names to other DNS servers for resolution.

Two forwarding modes are available: forward all requests without recursion and forward all requests with recursion.

- Forward all requests without recursion: Only a specified DNS server is used to resolve domain names. If the specified DNS server cannot resolve the domain names or the request times out, a message is returned to the DNS client to indicate that the query failed.
- Forward all requests with recursion: A specified DNS server is preferentially used to resolve domain names. If the specified DNS server cannot resolve the domain names, the local DNS is used instead.

29.1.4.2.1.2. View global forwarding domain names

This topic describes how to view global forwarding domain names in the Apsara Uni-manager Management Console. This operation requires administrator permissions.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domain Names > Forwarding Settings > Global**

Forwarding Domains.

3. In the **Domain Name** search box, enter the domain name that you want to query and click **Search**.

29.1.4.2.1.3. Add a domain name

This topic describes how to add a domain name in the Apsara Uni-manager Management Console. This operation requires administrator permissions.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domain Names > Forwarding Settings > Global Forwarding Domains**.
3. Click **Add Domain Name**.
4. In the dialog box that appears, configure *Global Forwarding Domain*, *Forwarding Mode*, and *Forwarder IP Addresses*. Then, click **OK**.

29.1.4.2.1.4. Add a description for a domain name

This topic describes how to add a description for a domain name in the Apsara Uni-manager Management Console. This operation requires administrator permissions.

Context

You can add a description for a domain name to help you easily identify it. For example, you can describe a domain name by using a hostname or internal system information.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domain Names > Forwarding Settings > Global Forwarding Domains**.
3. Select the domain name for which you want to add a description, click  in the Actions column, and then select **Description** from the shortcut menu.
4. In the dialog box that appears, enter a description and click **OK**.

29.1.4.2.1.5. Modify the forwarding configurations of a domain name

This topic describes how to modify the forwarding configurations of a domain name in the Apsara Uni-manager Management Console. This operation requires administrator permissions.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domain Names > Forwarding Settings > Global Forwarding Domains**.
3. Find the domain name whose forwarding configurations you want to modify, click the  icon in the Actions column, and then select **Modify**.
4. In the dialog box that appears, change the value of *Forwarding Mode* or *Forwarder IP Addresses*, and click **OK**.

29.1.4.2.1.6. Delete a domain name

This topic describes how to delete a domain name in the Apsara Uni-manager Management Console. This operation requires administrator permissions.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domain Names > Forwarding Settings > Global Forwarding Domains**.
3. Find the domain name that you want to delete, click the icon in the Actions column, and then select **Delete**.
4. In the message that appears, click **OK**.

29.1.4.2.1.7. Delete multiple domain names

This topic describes how to delete multiple domain names at the same time in the Apsara Uni-manager Management Console. This operation requires administrator permissions.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domain Domains > Forwarding Settings > Global Forwarding Domains**.
3. Select one or more domain names that you want to delete and click **Delete** in the upper-right corner of the domain name list.
4. In the message that appears, click **OK**.

29.1.4.2.2. Global default forwarding configurations

29.1.4.2.2.1. Enable default forwarding

This topic describes how to enable default forwarding in the Apsara Uni-manager Management Console. This operation requires administrator permissions.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domain Names > Forwarding Settings > Global Default Forwarding**.
3. Click the icon in the Actions column and select **Enable**.
4. In the dialog box that appears, configure *Default Forwarding Mode* and *Forwarder IP Addresses*. Then, click **OK**.

   Make sure that Enable Default Forwarding is set to ON.

29.1.4.2.2.2. Modify default forwarding configurations

This topic describes how to modify default forwarding configurations in the Apsara Uni-manager Management Console. This operation requires administrator permissions.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domain Names > Forwarding Settings > Global Default Forwarding**.
3. Click the icon in the Actions column and select **Modify**.
4. In the dialog box that appears, configure *Forwarding Mode* and *Forwarder IP Addresses*. Then, click **OK**.

29.1.4.2.2.3. Disable default forwarding

This topic describes how to disable default forwarding in the Apsara Uni-manager Management Console. This operation requires administrator permissions.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Domain Names > Forwarding Settings > Global Default Forwarding**.
3. Click the icon in the Actions column and select **Disable**.
4. In the message that appears, click **OK**.

29.1.4.3. Global recursive resolution

29.1.4.3.1. Enable global recursive resolution

Prerequisites

You have administrator permissions.

Procedure

1. Log on to the Apsara Stack DNS console.
2. Choose **Internal Domains > Global Recursive Resolution**.
3. Click the icon in the Actions column and select **Enable**.
4. In the dialog box that appears, click **OK**.

29.1.4.3.2. Disable global recursive resolution

Prerequisites

You have administrator permissions.

Procedure

1. Log on to the Apsara Stack DNS console.
2. Choose **Internal Domains > Global Recursive Resolution**.
3. Click the icon in the Actions column and select **Disable**.

4. In the dialog box that appears, click **OK**.

29.1.5. PrivateZone (DNS Standard Edition only)

The PrivateZone feature allows you to create VPC-specific tenant domain names. You can bind the domain names to VPCs as required to achieve tenant isolation.

29.1.5.1. Tenant internal domain name

29.1.5.1.1. View a domain name

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains**.
3. In the **Domain Name** search box, enter the domain name that you want to view.
4. Click Search.

   The search result is displayed.

29.1.5.1.2. Add a domain name

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains**.
3. Click **Add Domain Name**.
4. In the dialog box that appears, set *Tenant Internal Domain Name*.
5. Click **OK**.

29.1.5.1.3. Bind an organization to a VPC

Tenant domain names are isolated based on VPCs. To ensure that the DNS forwarding configurations take effect, you must bind the organization of domain names to a VPC.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains**.
3. Find the target domain name, click the  icon in the Actions column, and select **Associate VPCs**.
4. Select one or more VPCs from the list of VPCs to Select, click the right arrow to add them to the list of VPCs Selected, and then click **OK**.

29.1.5.1.4. Unbind a domain name from a VPC

This topic describes how to unbind a domain name from a VPC.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains**.

3. Find the target domain name and click the number in the **VPCs Associated** column.
4. On the **VPCs Associated** page, find the target VPC, click the icon in the **Actions** column, and then select **Disassociate**.

   Make sure that the unbound VPC is no longer displayed on the VPCs Associated page.

29.1.5.1.5. Add a description for a domain name

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains**.
3. Find the target domain name, click the icon in the Actions column, and then select **Description**.
4. In the dialog box that appears, enter a description.
5. Click **OK**.

29.1.5.1.6. Delete a domain name

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains**.
3. Find the target domain name, click the icon in the Actions column, and then select **Delete**.
4. In the message that appears, click **OK**.

29.1.5.1.7. Delete multiple domain names

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains**.
3. Select one or more domain names that you want to delete and click **Delete** in the upper-right corner.
4. In the message that appears, click **OK**.

29.1.5.1.8. Configure DNS records

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains**.
3. Find the target domain name, click the icon in the Actions column, and then select **Configure DNS Records**.
4. In the upper-right corner of the **Configure DNS Records** page, click **Add DNS Record**.
5. In the **Add DNS Record** dialog box, configure *Host*, *Type*, *TTL*, *Resolution Policy*, and *Record Set*. Then, click **OK**.

The following tables describe the types of DNS records.

- A record

Resolution policy	Formatting rule
None	You can enter up to 100 unique IPv4 addresses, each in a separate row. Make sure that the IPv4 addresses are valid. Example: ■ 192.168.1.1 ■ 192.168.1.2 ■ 192.168.1.3
Weight	You can enter up to 100 unique IPv4 addresses, each in a separate row. Format: ■ [IPv4 address] [Weight] (The IPv4 address and weight are separated with a space.) ■ Make sure that the IPv4 addresses are valid. ■ The weight value is an integer ranging from 0 to 999. A larger value indicates a greater weight. Example: ■ 192.168.1.1 20 ■ 192.168.1.1 30 ■ 192.168.1.1 50

- AAAA record

Resolution policy	Formatting rule
None	You can enter up to 100 unique IPv6 addresses, each in a separate row. Make sure that the IPv6 addresses are valid. Example: ■ 2400:3200::6666 ■ 2400:3200::6688 ■ 2400:3200::8888
Weight	You can enter up to 100 unique IPv6 addresses, each in a separate row. Format: ■ [IPv6 address] [Weight] (The IPv6 address and weight are separated with a space.) ■ Make sure that the IPv6 addresses are valid. ■ The weight value is an integer ranging from 0 to 999. A larger value indicates a greater weight. Example: ■ 2400:3200::6666 20 ■ 2400:3200::6688 20 ■ 2400:3200::8888 60

- CNAME record

Resolution policy	Formatting rule
None	You can enter only one domain name. The domain name must be a fully qualified domain name (FQDN) that ends with a dot (.). It must be 1 to 255 characters in length. Example: www.example.com.
Weight	You can enter up to 100 unique domain names, each in a separate row. Format: ■ [Domain name] [Weight] (The domain name and weight are separated with a space.) ■ The domain name must be an FQDN that ends with a period (.). It must be 1 to 255 characters in length. ■ The weight value is an integer ranging from 0 to 999. A larger value indicates a greater weight. Example: ■ www1.example.com. 20 ■ www2.example.com. 20 ■ www3.example.com. 60

- MX record

Resolution policy	Formatting rule
None	You can enter 100 unique email server hostnames, each in a separate row. Format: ■ [Priority] [Email server hostname] (The priority and hostname are separated with a space.) ■ The priority value is an integer ranging from 0 to 999. A smaller value indicates a higher priority. ■ The email server hostname must be an FQDN that ends with a period (.). It must be 1 to 255 characters in length. Example: ■ 10 mailserver1.example.com. ■ 20 mailserver2.example.com.

- TXT record

Resolution policy	Formatting rule
None	You can enter up to 100 unique character strings, each in a separate row. A string must be 1 to 255 characters in length. No row can be left blank. Example: "v=spf1 ip4:192.168.0.1/16 ip6:2001::1/96 ~all"

- PTR record

Resolution policy	Formatting rule
None	You can enter up to 100 unique domain names, each in a separate row. The DNS server address must be an FQDN that ends with a period (.). It must be 1 to 255 characters in length. Example: ■ www1.example.com. ■ www2.example.com. ■ www3.example.com.

- SRV record

Resolution policy	Formatting rule
None	You can enter up to 100 unique application server hostnames, each in a separate row. Format: ■ [Priority] [Weight] [Port number] [Application server hostname] (Every two items are separated with a space.) ■ The priority value is an integer ranging from 0 to 999. A smaller value indicates a higher priority. ■ The weight value is an integer ranging from 0 to 999. A larger value indicates a greater weight. ■ The port number is an integer ranging from 0 to 65535. It indicates the TCP or UDP port used for network communications. ■ The application server hostname must be an FQDN that ends with a period (.). It must be 1 to 255 characters in length. Example: ■ 1 10 8080 www1.example.com. ■ 2 20 8081 www2.example.com.

- NAPTR record

Resolution policy	Formatting rule

Resolution policy	Formatting rule
None	You can enter up to 100 unique NAPTR record values, each in a separate row. Format: ■ [Serial number] [Priority] [Flag] [Service information] [Regular expression] [Substitute domain name] (Every two items are separated with a space.) ■ The serial number is an integer ranging from 0 to 999. A smaller value indicates a higher priority. ■ The priority value is an integer ranging from 0 to 999. A smaller value indicates a higher priority. If two records have the same serial number, the one with a higher priority takes effect first. ■ The flag value can be left blank or be a character from A to Z, a to z, or 0 to 9. It is not case-sensitive and must be enclosed in double quotation marks (""). ■ The service information can be left blank or be a string of 1 to 32 characters. It must start with a letter and be enclosed in double quotation marks (""). ■ The regular expression can be left blank or be a string of 1 to 255 characters enclosed in double quotation marks (""). ■ The substitute domain name must be an FQDN that ends with a period (.). It must be 1 to 255 characters in length. Example: ■ 100 50 "S" "Z3950+I2L+I2C" "" _z3950._tcp.example.com. ■ 100 50 "S" "RCDS+I2C" "" _rcds._udp.example.com. ■ 100 50 "S" "HTTP+I2L+I2C+I2R" "" _http._tcp.example.com.

- CAA record

Resolution policy	Formatting rule
None	You can enter up to 100 unique CAA records, each in a separate row. Format: ■ [Certificate authority flag] [Certificate property tag] [Authorization information] (Every two items are separated with a space.) ■ The certification authority flag is an integer ranging from 0 to 255. ■ The certificate property tag can be issue, issuewild, or iodef. ■ The authorization information must be 1 to 255 characters in length and enclosed in double quotation marks (""). Example: ■ 0 issue "caa.example.com" ■ 0 issuewild ";" ■ 0 iodef "mailto:example@example.com"

- NS record

Resolution policy	Formatting rule

Resolution policy	Formatting rule
None	You can enter up to 100 unique DNS server addresses, each in a separate row. The DNS server address must be an FQDN that ends with a period (.). It must be 1 to 255 characters in length. Wildcard domain names are not allowed. Example: ■ ns1.example.com. ■ ns2.example.com.

6. After you add DNS records, perform the following operations as required:
   - Add a description for a DNS record.

     Find the target DNS record, click the icon in the Actions column, and then select **Description**. In the dialog box that appears, enter a description and click **OK**.
   - Delete a DNS record.

     Find the target DNS record, click the icon in the Actions column, and then select **Delete**. In the message that appears, click **OK**.
   - Modify a DNS record.

     Find the target DNS record, click the icon in the Actions column, and then select **Modify**. In the dialog box that appears, modify the required parameters and click **OK**.
   - Delete multiple DNS records.

     Select the DNS records that you want to modify and click **Delete** in the upper-right corner. In the message that appears, click **OK**.

29.1.5.1.9. View a resolution policy

This topic describes how to view the details of a resolution policy.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains**.
3. Find the target domain name, click the icon in the Actions column, and then select Configure DNS Records.
4. View the resolution policy in the DNS Records list.

29.1.5.2. Tenant forwarding configurations

29.1.5.2.1. Tenant forwarding domain names

29.1.5.2.1.1. View a tenant forwarding domain name

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains**.
3. In the **Domain Name** search box, enter the domain name that you want to view.
4. Click **Search**.

   The search result is displayed.

29.1.5.2.1.2. Add a tenant forwarding domain name

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains**.
3. Click **Add Domain Name**.
4. In the dialog box that appears, configure parameters such as *Domain Name*, *Forwarding Mode*, and *Forwarder IP Addresses*.

Parameter	Description
Domain Name	The domain name, which must meet the following formatting rules: ○ The domain name must be 1 to 255 characters in length. This includes the period (.) at the end of the domain name. ○ The domain name can contain multiple domain name segments that are separated with periods (.). A domain name segment must be 1 to 63 characters in length. It cannot contain consecutive periods (.) or be left blank. ○ The domain name can only contain letters (a to z, A to Z), digits (0 to 9), hyphens (-), and underscores (_). ○ The domain name must start with a letter, digit, or underscore (_) and end with a letter, digit, or period (.). ○ The domain name is not case-sensitive. The system saves the domain name in lowercase letters. ○ The period (.) at the end of the domain name is optional. The system adds a period (.) to the end of the domain name.
Forwarding Mode	For both domain name-based forwarding and default forwarding, the following two forwarding modes are supported: ○ Forward All Requests without Recursion: forwards DNS requests to the target DNS server. If the target DNS server cannot resolve the domain names, a message is returned to the DNS client indicating that the query failed. ○ Forward All Requests with Recursion: A specified DNS server is preferentially used to resolve domain names. If the specified DNS server cannot resolve the domain names, the local DNS is used instead. If you enter internal IP addresses in the Forwarder IP Addresses field, unexpected results may occur during recursive resolution. For example, a domain name used for internal network services may be resolved to a public IP address.
Forwarder IP Addresses	A list of destination IP addresses. **Note** Multiple IP addresses are separated with semicolons (;).

5. Click **OK**.

29.1.5.2.1.3. Bind an organization to a VPC

Tenant domain names are isolated based on VPCs. You must bind the organization of domain names to a VPC before the DNS forwarding configurations can take effect.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains**.
3. Find the target domain name, click the  icon in the Actions column, and then select **Associate VPCs**.
4. Select one or more VPCs from the list of VPCs to Select, click the right arrow to add them to the list of VPCs Selected, and then click **OK**.

29.1.5.2.1.4. Unbind a domain name from a VPC

This topic describes how to unbind a domain name from a VPC.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains**.
3. Find the target domain name and click the number in the **VPCs Associated** column.
4. On the VPCs Associated page, find the target VPC, click the  icon in the Actions column, and then select **Disassociate**.

   Make sure that the unbound VPC is no longer displayed on the VPCs Associated page.

29.1.5.2.1.5. Modify the forwarding configurations of a domain name

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains**.
3. Find the target domain name, click the  icon in the Actions column, and then select **Modify**.
4. In the dialog box that appears, change the value of **Forwarding Mode** or **Forwarder IP Addresses**.
5. Click **OK**.

29.1.5.2.1.6. Add a description for a tenant forwarding domain name

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains**.
3. Find the target domain name, click the  icon in the Actions column, and then select **Description**.
4. In the dialog box that appears, enter a description.
5. Click **OK**.

29.1.5.2.1.7. Delete a tenant forwarding domain name

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains**.
3. Find the target domain name, click the  icon in the Actions column, and then select **Delete**.
4. In the message that appears, click **OK**.

29.1.5.2.1.8. Delete multiple tenant forwarding domain names

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains**.
3. Select one or more domain names that you want to delete and click **Delete** in the upper-right corner.
4. In the message that appears, click **OK**.

29.1.5.2.2. Tenant default forwarding configurations

29.1.5.2.2.1. View default forwarding configurations

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding**.

29.1.5.2.2.2. Add a default forwarding configuration

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding**.
3. Click **Add Settings**.
4. In the dialog box that appears, configure parameters such as *Forwarding Mode* and *Forwarder IP Addresses*.

Parameter	Description
Forwarding Mode	For both domain name-based forwarding and default forwarding, the following two forwarding modes are available: ○ Forward All Requests without Recursion: Only a specified DNS server is used to resolve domain names. If the specified DNS server cannot resolve the domain names, a message is returned to the DNS client to indicate that the query failed. ○ Forward All Requests with Recursion: A specified DNS server is preferentially used to resolve domain names. If the specified DNS server cannot resolve the domain names, a local DNS server is used instead. If you enter internal IP addresses in the Forwarder IP Addresses field, unexpected results may occur during recursive resolution. For example, a domain name used for internal network services may be resolved to a public IP address.
Forwarder IP Addresses	A list of destination IP addresses. **Note** Multiple IP addresses are separated with semicolons (;).

5. Click **OK**.

29.1.5.2.2.3. Bind an organization to a VPC

Tenant domain names are isolated based on VPCs. You must bind the organization of domain names to a VPC before the DNS forwarding configurations can take effect.

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding**.
3. Find the target organization, click the  icon in the Actions column, and then select **Associate VPCs**.
4. Select one or more VPCs from the list of VPCs to Select, click the right arrow to add them to the list of VPCs Selected, and then click **OK**.

29.1.5.2.2.4. Unbind a domain name from a VPC

This topic describes how to unbind a domain name from a VPC.

Procedure

1. Log on to the Apsara Stack DNS console.

2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding**.
3. Find the target domain name and click the number in the **VPCs Associated** column.
4. On the VPCs Associated page, find the target VPC, click the icon in the Actions column, and then select **Disassociate**.

   Make sure that the unbound VPC is no longer displayed on the VPCs Associated page.

29.1.5.2.2.5. Modify a default forwarding configuration

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding**.
3. Find the target organization, click the icon in the Actions column, and then select **Modify**.
4. In the dialog box that appears, change the value of **Forwarding Mode** or **Forwarder IP Addresses**.
5. Click **OK**.

29.1.5.2.2.6. Add a default forwarding configuration

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding**.
3. Find the target organization, click the icon in the Actions column, and then select **Description**.
4. In the dialog box that appears, enter **Description**.
5. Click **OK**.

29.1.5.2.2.7. Delete a default forwarding configuration

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding**.

3. Find the target organization, click the  icon in the Actions column, and then select **Delete**.
4. In the dialog box that appears, click **OK**.

29.1.5.2.2.8. Delete multiple default forwarding configurations

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding**.
3. Select one or more domain names that you want to delete and click **Delete** in the upper-right corner.
4. In the message that appears, click **OK**.

29.1.6. Internal Global Traffic Manager (internal GTM Standard Edition only)

Internal Global Traffic Manager (GTM) supports multi-cloud disaster recovery for domain names of customers. This feature manages traffic loads between multiple Apsara Stack networks.

29.1.6.1. Scheduling instance management

29.1.6.1.1. Scheduling Instance

The Scheduling Instance tab displays all existing scheduling instances. You can add, delete, modify, and configure scheduling instances on this tab. When you create a scheduling instance, you must associate an address pool and scheduling domain with the instance.

29.1.6.1.1.1. Create a scheduling instance

After you create a scheduling instance, you can associate the scheduling instance with a scheduling domain and address pool.

1. In the left-side navigation pane, choose **Internal Global Traffic Manager** > **Scheduling Instances** > Scheduling Instance.

2. Click **Create Scheduling Instance** in the upper-right corner of the instance list.

3. In the dialog box that appears, configure Scheduling Instance Name, CNAME Access Domain Name, and Global TTL. Then, click **OK**.

29.1.6.1.1.2. Modify a scheduling instance

1. In the left-side navigation pane, choose **Internal Global Traffic Manager** > **Scheduling Instances** > Scheduling Instance.

2. Find the instance that you want to modify and click **Modify** in the Actions column.

3. Modify the parameter settings as prompted and click **OK**.

29.1.6.1.1.3. Configure a scheduling instance

You can add, delete, modify, and query access policies of scheduling instances.

Create an access policy for a scheduling instance

1. Log on to the Apsara Stack DNS console and choose Recursion Configurations > Scheduling Instances > Scheduling Instance. Find the target scheduling instance and click Configure in the Actions column. On the Access Policy Configuration page, click Create Access Policy in the upper-right corner of the page.

2. In the dialog box that appears, enter the required information and click OK.

Modify the access policy of a scheduling instance

1. On the Access Policy Configuration page, find the target access policy and click Modify in the Actions column.

2. In the dialog box that appears, modify the configurations and click OK.

Delete the access policy of a scheduling instance

1. On the Access Policy Configuration page, find the target access policy and click Delete in the Actions column.

2. In the dialog box that appears, click OK after you verify that the displayed information is correct.

29.1.6.1.1.4. Delete a scheduling instance

1. Log on to the Apsara Stack DNS console and choose Recursion Configurations > Scheduling Instances > Scheduling Instance.

2. Find the target instance and click Delete in the Actions column.

3. In the dialog box that appears, click OK.

Note: After you delete the instance, its configuration data is also deleted.

29.1.6.1.2. Address Pool

The Address Pool tab allows you to manage address pools. You can associate address pools with scheduling instances. The address pools are classified into three types: IPv4 address pool, IPv6 address pool, and domain name address pool. The load balancing policy of an address pool can be set to polling or weight.

29.1.6.1.2.1. Create an address pool

1. In the left-side navigation pane, choose **Internal Global Traffic Manager > Scheduling Instances > Address Pool**.

2. Click **Create Address Pool** in the upper-right corner of the address pool list.

3. Configure **Address Pool Name**, **Address Type**, **Load Balancing Policy**, and **Address List**. Then, click **OK**.

29.1.6.1.2.2. Modify the configurations of an address pool

1. In the left-side navigation pane, choose **Internal Global Traffic Manager > Scheduling Instances > Address Pool**.

2. Find the address pool whose configurations you want to modify and click **Modify** in the Actions column.

3. In the dialog box that appears, modify the configurations as required and click **OK**.

29.1.6.1.2.3. Delete an address pool

1. Log on to the Apsara Stack DNS console and choose Recursion Configurations > Scheduling Instances > Address Pool.

2. Find the target address pool and click Delete in the Actions column.

3. In the dialog box that appears, click OK after you verify that the displayed information is correct.

29.1.6.1.3. Scheduling Domain

The Scheduling Domain tab allows you to add, delete, and query scheduling domains.

You can log on to the Apsara Stack DNS console and choose Internal Global Traffic Manager > Scheduling Instances > Scheduling Domain to go to the scheduling domain list.

29.1.6.1.3.1. Create a scheduling domain

1. Log on to the Apsara Stack DNS console and choose Recursion Configurations > Scheduling Instances > Scheduling Domain. Then, click Create Scheduling Domain in the upper-right corner of the scheduling domain list.

2. In the dialog box that appears, enter the custom domain name and click OK.

29.1.6.1.3.2. Add a description for a scheduling domain

1. In the left-side navigation pane, choose Internal Global Traffic Manager > Scheduling Instances > Scheduling Domain.

2. Find the scheduling domain for which you want to add a description and click Edit in the Actions column.

3. In the dialog box that appears, add a description in the Edit field and click OK.

29.1.6.1.3.3. Delete a scheduling domain

1. In the left-side navigation pane, choose Internal Global Traffic Manager > Scheduling Instances > Scheduling Domain.

2. Find the scheduling domain that you want to delete and click Delete in the Actions column.

3. In the message that appears, click OK after you verify that the displayed information is correct.

29.1.6.2. Scheduling line management

29.1.6.2.1. IP Address Line Configuration

The IP Address Line Configuration tab allows you to define lines based on IP addresses. The lines are used to group request sources to achieve intelligent load balancing.

29.1.6.2.1.1. Add a line

1. In the left-side navigation pane, choose **Internal Global Traffic Manager** > **Scheduling Line Management** > IP Address Line Configuration.

2. Click **Add Line** in the upper-right corner of the line list.

3. In the dialog box that appears, configure the parameters as prompted and click **OK**.

29.1.6.2.1.2. Sort lines

1. In the left-side navigation pane, choose **Internal Global Traffic Manager** > **Scheduling Line Management** > IP Address Line Configuration.

2. Find the line whose sequence you want to change and click **Sort** in the Actions column.

3. Specify Sort Behavior as prompted and click **OK**.

29.1.6.2.1.3. Modify the configurations of a line

1. In the left-side navigation pane, choose **Internal Global Traffic Manager** > **Scheduling Line Management** > IP Address Line Configuration.
2. Find the line whose configurations you want to modify and click **Modify** in the Actions column.
3. Modify the configurations as prompted and click **OK**.

29.1.6.2.1.4. Delete a line

1. In the left-side navigation pane, choose **Internal Global Traffic Manager** > **Scheduling Line Management** > **IP Address Line Configuration**.
2. Find the line that you want to delete and click **Delete** in the Actions column.
3. In the message that appears, click **OK**.

29.1.6.3. Data synchronization management

Data synchronization management is used to synchronize Global Traffic Manager (GTM) data between clouds.

29.1.6.3.1. Synchronization cluster management

Synchronization clusters involve two operations: **Set Emergency Group** and **Merge GTM Control Domain**.

You can perform the following operations to go to the synchronization cluster management page:

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose **Internal Global Traffic Manager**.
3. On the page that appears, click the **Data Synchronization** tab.

Set Emergency Group

You can select some service instances to form a cluster to provide services.

- Enable the emergency group feature: If the synchronization cluster is abnormal, click **Set Emergency Group**. In the Set Emergency Group dialog box, turn on Emergency Group Switch, select available service instances to form an emergency group, and then click **OK**.
- Disable the emergency group feature: If the synchronization cluster is restored to normal, click **Set Emergency Group**. In the Set Emergency Group dialog box, turn off the Emergency Group Switch and click **OK**.

Merge GTM Control Domain

In multi-cloud scenarios, you can click **Merge GTM Control Domain** and enter the IP address of the leader service instance of the merged Global Traffic Manager (GTM) control domain to form a large synchronization cluster.

View the status of the synchronization cluster

You can view the status of the synchronization cluster on the Synchronization Cluster Management tab.

View the service instances in the synchronization cluster

You can view the following information of the service instances in the current synchronization cluster:

Instance IP Address, Instance Role, Status, Latest Synchronization Log ID, IP Address, and Instance Description.

You can also perform the following operations to switch the role of a service instance in the synchronization cluster from follower to leader:

1. Find the service instance with the follower role and click **Switch Primary** in the Actions column.

2. In the message that appears, click **OK**.