ALIBABA CLOUD

Alibaba Cloud

Apsara Stack Enterprise

Technical Whitepaper

Product Version: 2105, Internal: V3.14.0

Document Version: 20210826

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.IDC requirements

The features and performance of Apsara Stack platforms and services depend on the reliability (24/7 stable operation of servers and network devices) of Apsara stack data centers. This stability relies on the reliability of a series of complex infrastructure such as cooling and power supply. We recommend that you abide to tier 3 or a similar classification when building data centers that host Apsara Stack platforms to reduce stability risks in essence.

1.1. Environment requirements

This topic describes the environment requirements for Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Areas prone to flooding, such as the downstream of dams or flood-prone regions	Data centers cannot be set up in such areas.	Required
2	Areas prone to landslides, debris flows, or mountain slopes	Data centers cannot be set up in such areas.	Required
3	Seismic zones or fault zones	Data centers cannot be set up in such areas.	Required
4	Distance from areas where have experienced 100-year floods	No less than 100 meters.	Required
5	Distance from hazardous areas in chemical plants, landfills, gas stations, and polluted sites that have flammables and explosives such as dangerous chemicals and gas.	No less than 400 meters.	Required
6	Distance from military arsenals	No less than 1,600 meters.	Required
7	Distance from airports	The distance from both sides of the runway is no less than 1,000 meters. The distance from runways in the direction of takeoff and landing is no less than 8,000 meters.	Required
8	Distance from public parking lots	No less than 20 meters.	Required
9	Main roads of the physical park	At least two roads are required. One road must be a two-lane, two-way road, which can accommodate trucks 15 meters long and 3 meters wide.	Recommended
10	Distance from commercial and residential areas	No greater than 16,000 meters.	Recommended

No.	Description	Requirement	Matching type
11	Physical park	The physical park is independent or can be isolated to provide secure isolation.	Recommended

1.2. Building requirements

This topic describes the building requirements for Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Gross floor area of a single building	No less than 8,000 square meters.	Recommended
2	Acceptance of fire protection systems installed in buildings	Fire protection systems installed in buildings are tested and approved by the local fire department.	Required
3	Floor load capacity	More than 1,000 kg per square meters.	Required
4	Layer height	The clear span of buildings is greater than 3.6 meters.	Required
5	Transportation	Freight elevators are required for buildings no less than two floors and have a weight capacity of no less than two tons. The transportation aisles are no less than 2.4 meters wide and no less than 2.5 meters high.	Required
6	Classification of seismic protection of building constructions	The classification of seismic protection of building constructions is not lower than building type C.	Required
7	Fire-resistance rating	No less than Level 2.	Required
8	Waterproof rating	Level 1.	Required

1.3. Power system

This topic describes the requirements for power systems in Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Power introduction	At least a written certificate with power supply assurance is required.	Required

No.	Description	Requirement	Matching type
2	Route requirements for mains supply introduction	Dual routes are required and their distance must be greater than 10 meters. Cables are routed to the park on different roads.	Required
3	Requirements for mains supply introduction to substations	Class-A mains supply and two different circuits or two 10 kV, 35 kV, or 110 kV substations are used.	Required
4	Diesel generators	Diesel generators are configured for N + 1 redundancy. Diesel generators can start under load within two minutes.	Required
5	Period of time for which oil in tanks can be used	Greater than eight hours.	Required
6	Uninterruptible power supply (UPS) and redundancy	A UPS system based on 2N redundancy configuration is used for AC distribution. Or a high-voltage direct current (HVDC) system is used for a single mains supply.	Required
7	Period of time for which storage batteries can be discharged	No less than 15 minutes.	Required
8	Cabinet power distribution	A dual-circuit power supply system is used, which includes transformers, distribution lines, uninterruptible power supply, rack-mountable power distribution cabinets, and rack power distribution units (PDUs).	Required
9	Cabinet power consumption	No less than eight kW.	Recommended

1.4. Cooling system

This topic describes the requirements for the cooling system in Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Air conditioners, water pumps, water chiller units, and cooling towers in data centers	Air conditioners, water pumps, water chiller units, and cooling towers in data centers are configured for N + 1 redundancy.	Required
2	Power distribution for precision air conditioners in a chilled water system	Uninterrupted power supply (UPS)	Required
3	Power distribution for water supply pumps in a water-cooling system	UPS	Required
4	Period of time for which cool storage equipment can provide cooling	Time period during which cool storage equipment can provide cooling is no less than 10 minutes. When the cooling system is interrupted, the temperature of cold aisles in data centers cannot exceed 30 degrees Celsius.	Required
5	Building automation system	UPS. Redundancy must be provided for the direct digital controller (DDC) system and servers.	Recommended

1.5. Monitoring requirements

This topic describes the monitoring requirements for Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Monitoring access standards	Network communication is enabled based on TCP or IP sockets.	Recommended
2	Monitoring scope	The following items in data centers are monitored: temperature and humidity inside the data centers, terminal devices of the air conditioning system, chillers, pumps of the air conditioning system, power distribution cabinets, high-voltage direct current (HVDC) systems, uninterrupted power supply (UPS) systems, transformers, diesel generators, and mains supply.	Required

1.6. O&M requirements

This topic describes the O&M requirements for Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Technical team	A technical team must consist of the following personnel: one person for building decoration, one to two persons for air conditioning and refrigeration, one to two persons for high voltage power system, and at least one person for low voltage system monitoring.	Recommended
2	Construction delivery capability	The business deployment requirements are met (1,000 cabinets delivered within six months).	Recommended
3	Service-level agreement (SLA)	The availability of power, cooling, and network is above 99.99%.	Recommended
4	O&M personnel in the O&M system	The level and number of O&M personnel are confirmed.	Required
5	Professional qualifications of O&M personnel in the O&M system	The number of professional and technical personnel is no less than two in each of the following fields: electrical system, heating, ventilation, and air conditioning (HVAC), fire protection, and low voltage system.	Recommended
6	Duty system of the O&M system	The personnel on duty and emergency response mechanism are available 24/7/365 for infrastructure and network maintenance in data centers.	Required
7	Hardware and software maintenance of devices in the O&M system	A 24/7/365 professional maintenance service is purchased.	Required
8	Building management system (BMS) and video surveillance in the O&M system	The power and environment supervision system or BMS is used to monitor the running status of key infrastructure. The 24/7 video surveillance is provided, and the records are retained for 90 days.	Required

No.	Description	Requirement	Matching type
9	Entry and exit management of personnel and articles in the O&M system	A clear management process is provided, and records are complete and traceable.	Required
10	Service qualification	IDC business qualification: An Internet Data Centre Value Added Telecom Service license (IDC VATS) issued by the Chinese government is recommended.	Recommended
11	Third-party certification	The SSAE 16, ISO 17799, and ISO 9001 audits are passed. SSAE 16 ensures that service providers have sufficient security controls and safeguards in place to protect the security of user data. ISO 17799 ensures that the information security of service providers is less likely to be damaged. ISO 9001 sets out the criteria for a quality management system (QMS) of service providers.	Recommended
12	Operator personnel handover interface	A clear personnel handover interface is provided, including the assignment of roles and responsibilities, and the problem escalation path to the personnel who is in charge of the project and who holds the highest rank on the operator side. All responsibilities must be assigned and confirmed at the beginning of the project and be carried out for the entire project.	Recommended

1.7. Communication requirements

This topic describes the communication requirements for Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Number of direct routes between two data centers	Two direct routes with distances greater than 500 meters. Cables cannot be routed on the same conduit, trench, or route.	Required

No.	Description	Requirement	Matching type
2	Number of outgoing routes in a single data center	Provide two outgoing routes. The number of outgoing routes can be expanded to three as required before the project is delivered.	Recommended
3	Number of outgoing optical fibers	No less than 20 pairs.	Recommended
4	Routing method of optical cables	Optical cables must be placed into buried conduits. Overhead cabling is not allowed.	Required
5	Connection method of optical cables	Optical cables must be connected inside data centers. Outdoor connections are not allowed.	Required
6	Outgoing conduits	The park has more than two outgoing conduits in different directions and their distance is greater than 50 meters. Each outgoing conduit corresponds to a different entrance room.	Recommended
7	Communication rooms	There are two separate communication rooms in each data center.	Recommended
8	Leased lines and bandwidth	The data centers have the capabilities to support leased lines, Border Gateway Protocol (BGP) lines, and static bandwidth.	Recommended
9	Optical cable routes inside data centers	Cables inside data centers must be routed separately to ensure dual routes. The distance between the two routes must be greater than 10 meters.	Required
10	Access to optical cables of other operators	The data centers can access to optical cables of other operators.	Recommended

No.	Description	Requirement	Matching type
11	Number of direct routes between buildings	Two routes are required and four routes are preferred within physical fences. Three routes are required and four routes are preferred outside fences built on property lines. These routes can be completed when data centers are delivered. Direction of the routes must be approved by Alibaba Cloud.	Recommended
12	Number of optical fibers between buildings	At least 384-core × 4 optical fibers are required and can be scaled out.	Recommended

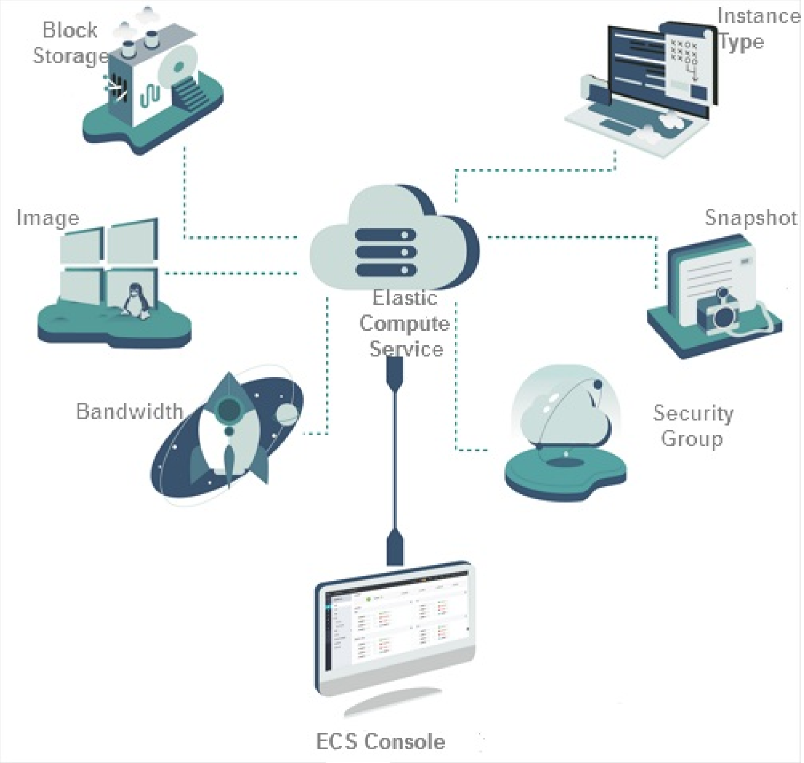
2.Elastic Compute Service (ECS)

2.1. What is ECS?

Elastic Compute Service (ECS) is a computing service that features elastic processing capabilities. Compared with physical servers, ECS instances are more user-friendly and can be managed more efficiently. 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 through the ECS console. Other resources, such as block storage, images, and snapshots, can only be used after they are integrated with ECS instances. For more information, see ECS components.

ECS components



2.2. Architecture

2.2.1. Overview

The ECS system is composed of a virtualization platform with distributed storage, a control system, and an O&M and monitoring system.

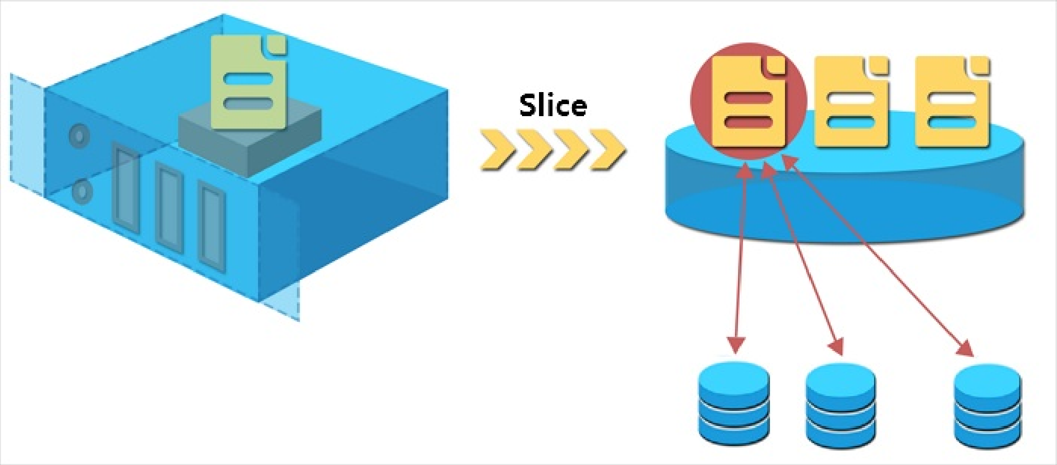
2.2.2. Virtualization platform and distributed storage

Virtualization is the foundation of ECS instances. Apsara Stack uses the Kernel-based Virtual Machine (KVM) virtualization solution to virtualize physical resources and provide them as ECS resources.

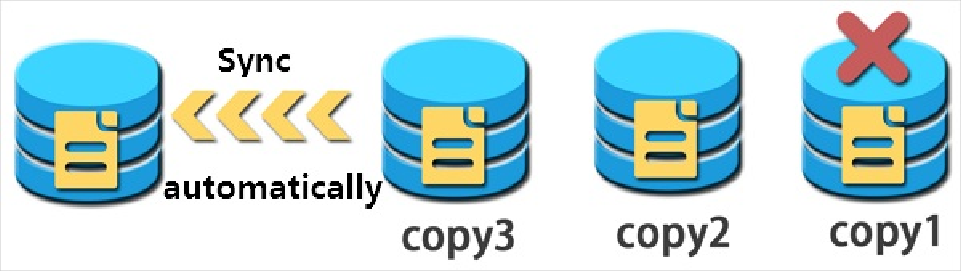
An ECS instance contains two important modules: the computing resource module and the storage resource module.

- Computing resources refer to CPU, memory, and bandwidth resources. These resources are created by virtualizing the resources of a physical server and then allocating them to ECS instances for use. The computing resources of a single ECS instance are based on those of a single physical server. When the resources of that physical server are exhausted, you must create new ECS instances on another physical server to obtain more resources. Resource Quality of Service (QoS) ensures that different ECS instances on a single physical server do not conflict with each other.
- ECS storage is provided by a large-scale distributed storage system. The storage resources of an entire cluster are virtualized and integrated into an external service. The data of a single ECS instance is distributed throughout the entire cluster. In the distributed storage system, all data is saved in triplicate. This allows damaged data in one copy to be automatically replicated from the other copies.

Triplicate backup



Automatic replication



2.2.3. Control system

The control system is the core of ECS. It determines the physical server on which to start ECS instances as well as processes and maintains all of the features and information of the ECS instances in a centralized manner.

The control system consists of the following modules:

- **Data collection module**

This module is responsible for collecting data throughout the virtualization platform, including data about the usage of computing, storage, and network resources. The data collection module serves as the basis for resource scheduling and allows you to perform centralized monitoring and management of cluster resource usage.

- Resource scheduling system

  This module determines on which physical server to start ECS instances. When an ECS instance is created, this module schedules the ECS instance based on the resource loads of the physical server. This module also determines where to restart an ECS instance when the instance fails.

- ECS management module

  This module manages and controls ECS instances such as starting, stopping, or restarting instances.

- Security control module

  This module monitors and manages the network security of the entire cluster.

2.2.4. ECS Bare Metal Instance

ECS Bare Metal Instance is a compute service that combines the elasticity of virtual machines and the performance and features of physical machines. ECS Bare Metal Instance is designed based on the state-of-the-art virtualization 2.0 technology developed by Alibaba Cloud. The virtualization technology used by ECS Bare Metal Instance is optimized to support common ECS instances and nested virtualization. It maintains the elastic performance of ECS instances and the performance and features of physical machines.

ECS Bare Metal Instance combines the strengths of both physical machines and ECS instances to deliver powerful and robust computing capabilities. ECS Bare Metal Instance uses virtualization 2.0 to provide your business applications with direct access to the processor and memory resources of the underlying servers without virtualization overheads. ECS Bare Metal Instance retains the hardware feature sets (such as Intel® VT-x) and resource isolation capabilities of physical machines, which is ideal for applications that need to run in non-virtualization environments.

By virtue of the independently developed chips, hypervisor system software, and the redefined server hardware architecture, ECS Bare Metal Instance integrates features from both physical and virtual machines. ECS Bare Metal Instance can seamlessly connect with other Apsara Stack services for storage, networking, and database tasks. ECS Bare Metal Instance is fully compatible with ECS instance images. These properties allow you to build resources to suit your business requirements.

When you use ECS Bare Metal Instance, take note of the following items:

- ECS Bare Metal Instance does not support instance type changes.
- When the physical machine that hosts an ECS bare metal instance fails, the system fails the instance over to another physical machine. Data is retained within the data disks of the instance.

2.3. Features

This topic describes the features of ECS instances.

ECS instances are the core components that provide computing services to users in ECS. It takes only a few minutes to create and start an ECS instance. When an ECS instance is created, it has specific system configurations. ECS instances allow you to compute business data more efficiently than traditional servers.

ECS instances are used and managed in the same manner as physical servers. You can perform a series of basic operations on ECS instances remotely or by calling API operations.

The processing power of ECS instances can be expressed in terms of virtual CPUs and virtual memory, while the storage capabilities of ECS disks are measured by the available capacity of cloud disks. ECS instances support more flexible machine configurations than traditional servers. If you find that the configurations of an ECS instance do not meet your business needs, you can change them at any time.

The lifecycle of an ECS instance begins when it is created and ends when it is released. After an ECS instance is released, all of its data is permanently deleted and cannot be recovered.

The ECS console consists of the following pages:

- Overview

  You can view the number of created and running instances as well as the distribution of ECS resources in each zone.

- Instances

  On the Instances page, you can view and manage your created instances. You can start, stop, restart, and release instances, as well as log on to the VNC management terminal, replace system disks, modify passwords, and change instance configurations. You can also view the basic information and configurations of instances.

- Disks

  On the Disks page, you can view and manage your created disks. You can re-initialize disks online, create snapshots, configure automatic snapshot policies, release disks, and attach or detach disks. You can also view the basic information and attaching information of disks.

- Images

  On the Images page, you can view and manage your created or shared images. You can copy, share, and delete images.

- Snapshots

  On the Snapshots page, you can view and manage your created snapshots. You can restore disks online, create custom images, and delete snapshots.

- Automatic snapshot policies

  On the Automatic Snapshot Policies page, you can view and manage your created automatic snapshot policies. You can batch configure automatic snapshot policies, modify automatic snapshot policy information, and delete automatic snapshot policies.

- Security groups

  On the Security Groups page, you can view and manage your created security groups. You can create, modify, delete, and batch delete security groups, as well as view the instances and rules associated with a security group.

- ENIs

  On the ENIs page, you can view and manage your created elastic network interfaces (ENIs). You can create, modify, and delete ENIs, as well as bind ENIs to or unbind ENIs from ECS instances.

- Deployment sets

  On the Deployment Sets page, you can view and manage your created deployment sets. You can create, modify, and delete deployment sets, as well as view the basic information of deployment sets.

3.Container Service for Kubernetes

3.1. What is Container Service?

Container Service provides high-performance, enterprise-class management for scalable Kubernetes-based containerized applications throughout the application lifecycle.

Container Service simplifies the creation and scaling of container management clusters. It integrates Apsara Stack virtualization, storage, network, and security capabilities, providing the optimal environment to run Kubernetes-based containerized applications in the cloud. Alibaba Cloud is a Kubernetes certified service provider, with Container Service being among the first services to pass the Certified Kubernetes Conformance Program. Container Service provides professional container support and services.

3.2. Container technology

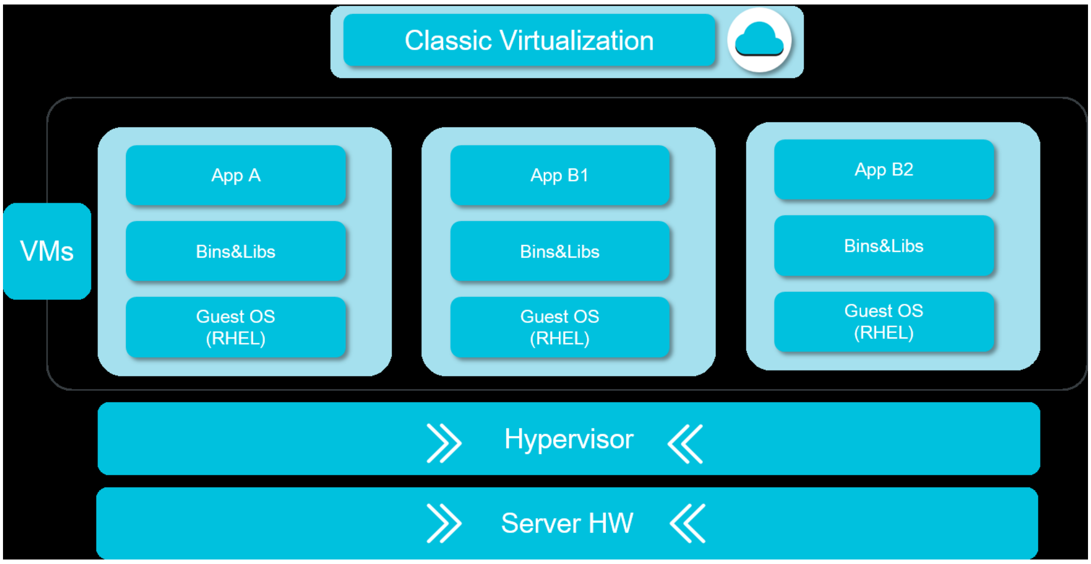
Containers are a lightweight operating system-level virtualization technology. You can use container images to deliver applications. Container images include applications and their necessary runtime dependencies. Container images have excellent portability and ensure deployment consistency in different environments. Containers are isolated from each other during runtime, ensuring excellent security.

Containers avoid potential version conflicts resulting from different applications running in the same environment, and eliminate runtime environment inconsistencies resulting from the same software being run in different environments. Because all containers on a host share the host's OS kernel, containers are more lightweight than virtual machines. This allows you to start containers quickly and gain fine-grained control over container resources.

Container technology and virtualization

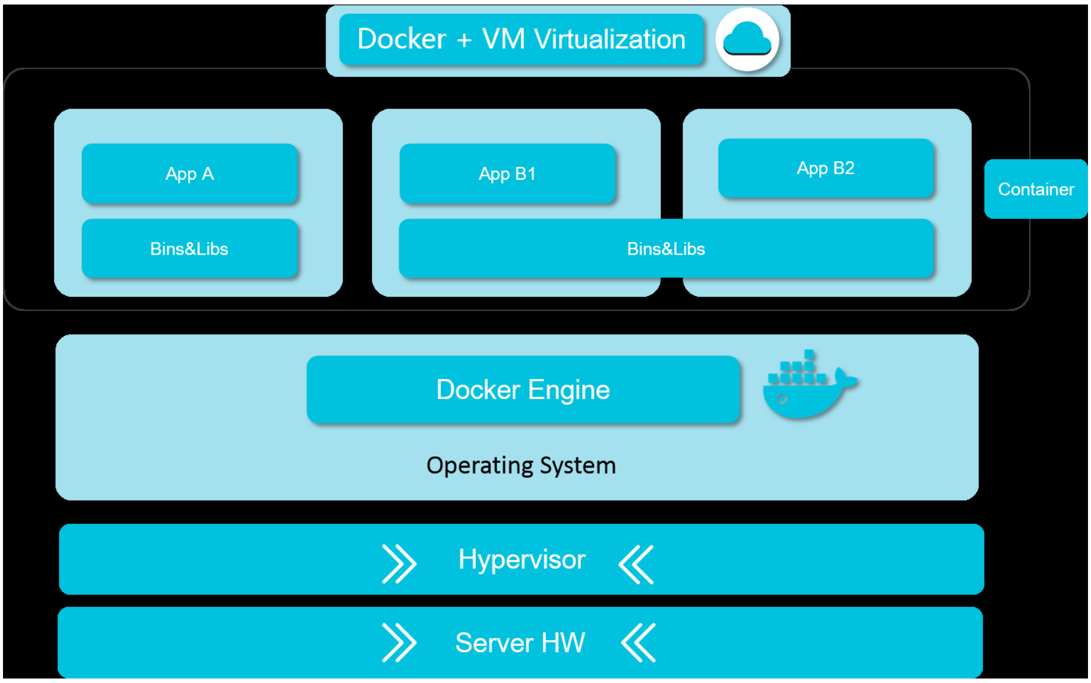
Containers do not conflict with conventional virtualization technologies. Conventional virtualization technologies encompass all elements ranging from operating systems to applications, as shown in the following figure.

Classic virtualization



Containers only package the application code and its runtime environments. Images can be reused within the same environment in different containers, making containers simple to use and operate.

Combination of Docker and virtualization



By combining containers and virtualization technologies, you can use virtual machines to provide an elastic infrastructure that offers improved security isolation and live migration capabilities. You can also use the container technology to streamline the deployment and O&M of applications and implement an elastic application architecture.

Technical features

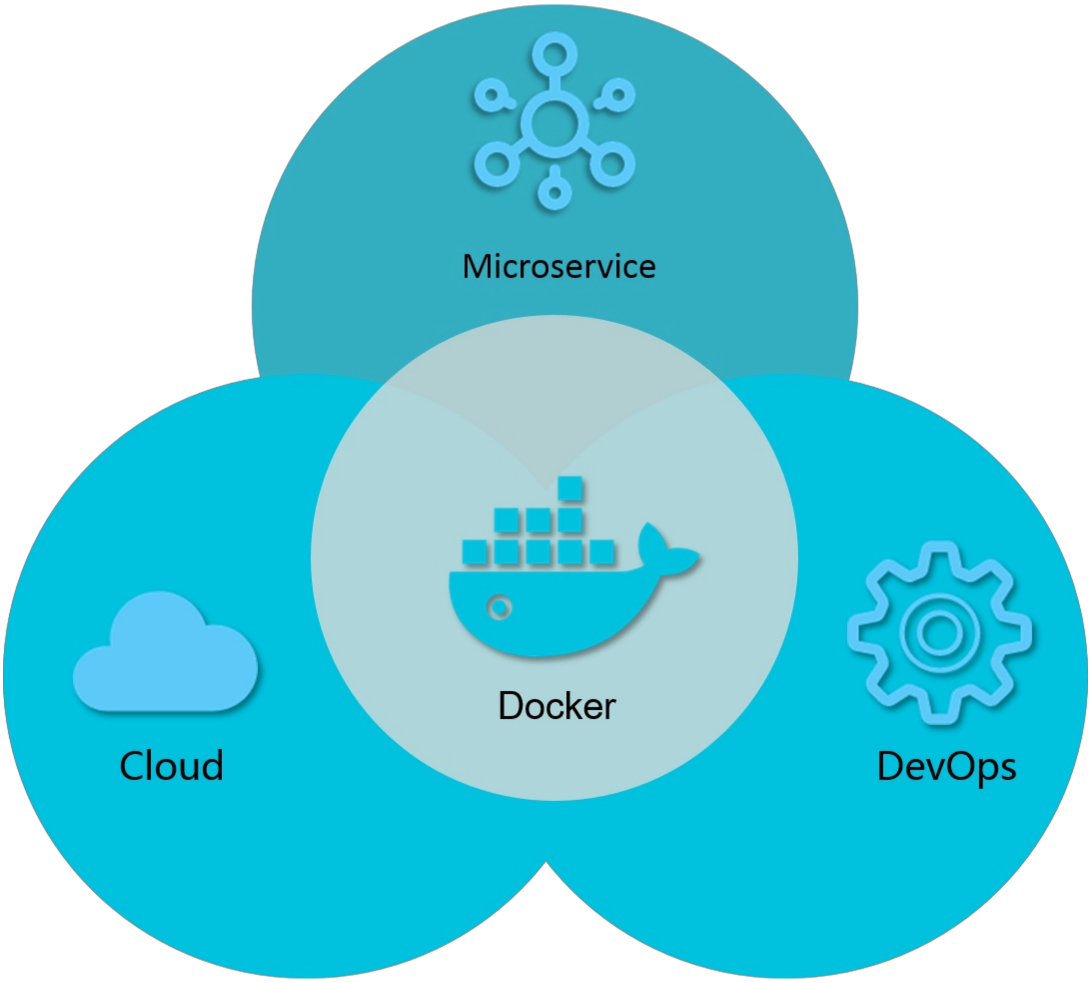
Containers are agile, portable, and highly-controllable.

- **Agility**: Containers attract developers with their simplicity and velocity, and allow enterprises to consistently develop and deliver software with greater efficiency.
- **Portability**: Developers can migrate containerized applications from the development environment, to the testing environment, and ultimately to the production environment. During this process, the operating structures for identical images are consistent. Computing capabilities can be deployed across data centers, making computing capability migration a reality in hybrid clouds.
- **Controllability**: Applications in the production environment must meet SLA goals. This requires that you have comprehensive management, security, and monitoring capabilities. Containers provide standardized application environments, allowing developers to use automated tools to manage the infrastructures and applications and ensure that all operations are automated, controllable, and traceable.

Scenarios

Containers can be applied in a wide range of scenarios. Containers are most often discussed and researched in relation to scenarios that have high container technology requirements, especially DevOps, cloud application management, and microservices.

Common scenarios



3.3. Architecture

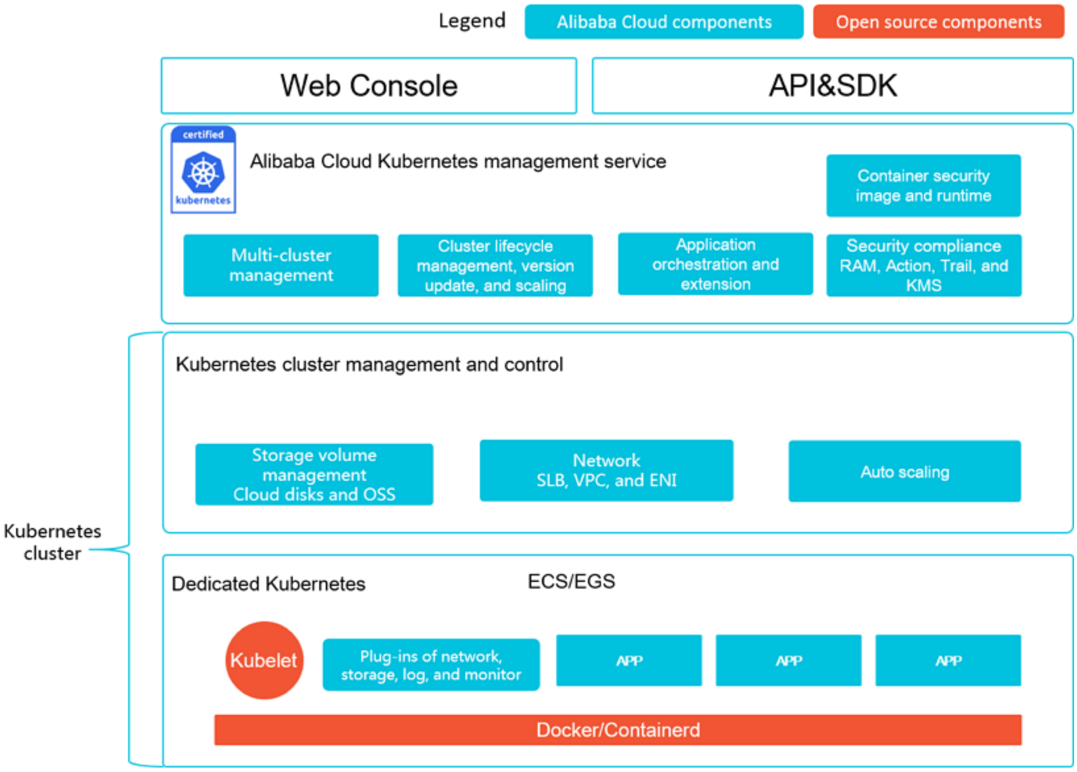
Apsara Stack Container Service supports YAML orchestration and cluster management for Kubernetes to extend and optimize third-party capabilities on Apsara Stack. Container Service allows you to manage clusters and containerized applications through GUIs and APIs.

The underlying architecture allows you to use exclusive cloud servers or physical servers to create a secure and controllable underlying environment where you can customize security group and VPC security rules.

To help migrate your applications to the cloud at a lower cost, Container Service implements APIs that are compatible with standard Docker APIs and all Docker images. Container Service provides Kubernetes YAML orchestration templates which allow you to migrate your applications seamlessly to the cloud. It also provides flexible and customizable mechanisms for third-party capability extensions.

The following figure shows the Container Service architecture.

Architecture

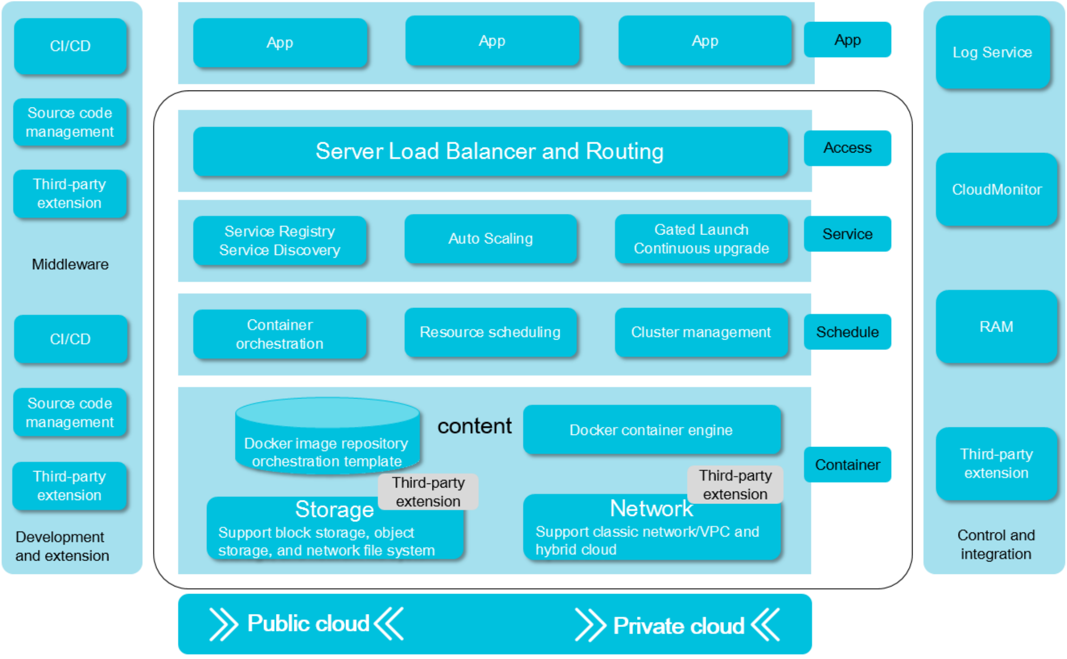

Legend
Alibaba Cloud components
Open source components
Web Console
API&SDK
certified
kubernetes
Alibaba Cloud Kubernetes management service
Container security image and runtime
Multi-cluster management
Cluster lifecycle management, version update, and scaling
Application orchestration and extension
Security compliance RAM, Action, Trail, and KMS
Kubernetes cluster management and control
Storage volume management Cloud disks and OSS
Network SLB, VPC, and ENI
Auto scaling
Kubernetes cluster
Dedicated Kubernetes
ECS/EGS
Kubelet
Plug-ins of network, storage, log, and monitor
APP
APP
APP
Docker/Containerd

Container Service is adapted and enhanced on the basis of native Kubernetes. This service simplifies cluster creation and scaling and integrates Apsara Stack virtualization, storage, network, and security capabilities, providing the optimal environment to run Kubernetes-based containerized applications in the cloud.

Feature	Description
Dedicated Kubernetes mode	Integrated with Apsara Stack virtualization technologies, the service allows you to create dedicated Kubernetes clusters. Elastic Compute Service (ECS), Elastic GPU Service (EGS), and ECS Bare Metal instances can be used as cluster nodes. Instances can be flexibly configured to different specifications and support a wide range of plug-ins.
Apsara Stack Kubernetes cluster management and control service	The service provides powerful network, storage, cluster management, scaling, and application extension features.
Apsara Stack Kubernetes management service	The service supports secure images and is highly integrated with Apsara Stack Resource Access Management (RAM), Key Management Service (KMS), and logging and monitoring services to provide a secure and compliant Kubernetes solution.
Convenient and efficient use	Container Service for Kubernetes provides services through the Web console, APIs and SDKs.

The following figure shows the Container Service capability stack. Container Service is built on a cloud infrastructure. It is deeply integrated with Apsara Stack capabilities, and supports third-party extensions and applications.

Functional architecture



3.4. Features

Features

Cluster management

- With the Container Service console, you can easily create a classic dedicated Kubernetes cluster supporting GPU servers within 10 minutes.
- Provides container-optimized OS images as well as Kubernetes and Docker versions that have undergone **stability testing and security enhancement**.
- Supports multi-cluster management, cluster upgrades, and cluster scaling.

Provides end-to-end container lifecycle management

- **Network**

  Provides high performance VPC and elastic network interface (ENI) plug-ins optimized for Apsara Stack, boasting 20% increased performance compared with regular network solutions.

  Supports container access and throttling policies.
- **Storage**

  Container Service is integrated with Apsara Stack disks and OSS, and provides the standard FlexVolume drive.

  Supports real-time creation and migration of volumes.
- **Logs**

  Provides high-performance log collection integrated with Apsara Stack Log Service.

Supports the integration with third-party open-source logging solutions.

- **Monitoring**

  Supports both container-level and VM-level monitoring. Integration with third-party open-source monitoring solutions is supported.

- **Permissions**

  Supports cluster-level Resource Access Management (RAM).

  Supports application-level permission configuration management.

- **Application management**

  Supports phased release and blue-green release.

  Supports application monitoring and scaling.

High-availability scheduling policies that allow you to easily handle upstream and downstream delivery processes

- Supports service-level affinity policies and scale-out.
- Provides high availability and disaster recovery across zones.
- Provides cluster and application management APIs to easily implement continuous integration and private system deployment.

4.Auto Scaling (ESS)

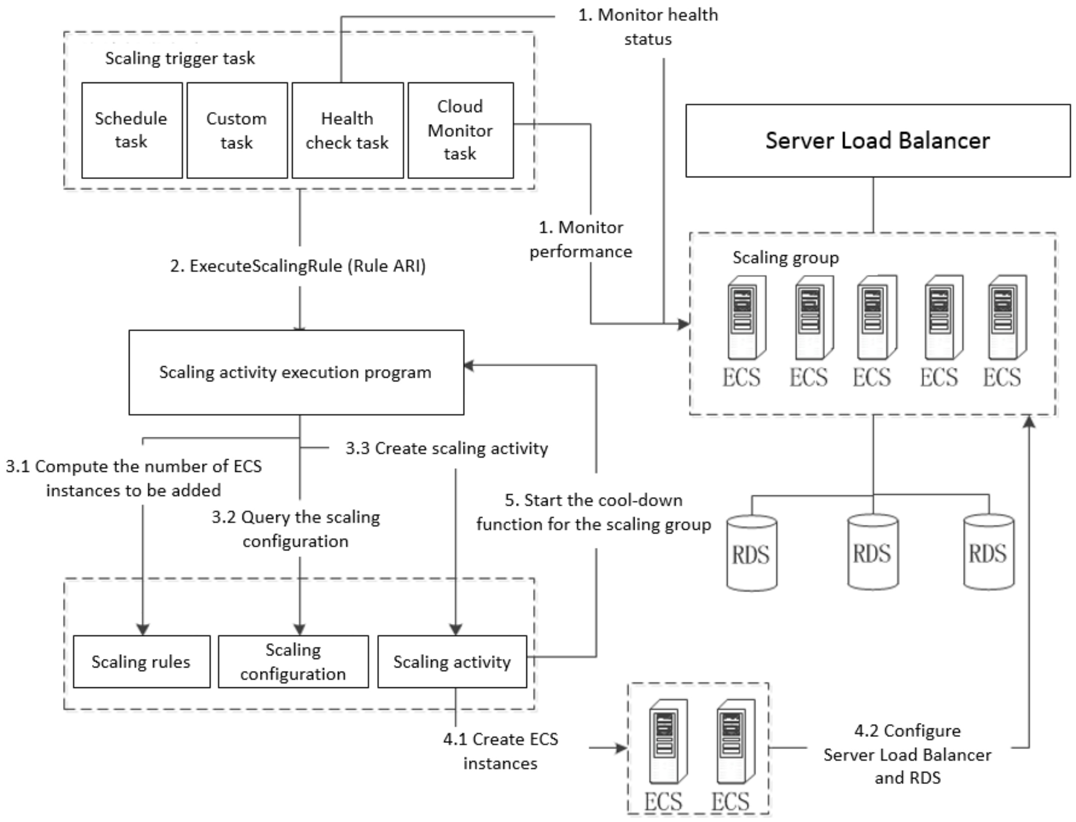
4.1. What is Auto Scaling?

Auto Scaling is a management service that automatically adjusts the number of elastic computing resources based on your business requirements and policies. It is suitable for applications with fluctuating or stable business loads.

Auto Scaling automatically schedules computing resources based on customer policies and business changes. It provides support for changing business loads and helps control infrastructure costs within an acceptable range. Auto Scaling automatically creates ECS instances based on user-defined scaling policies and modes. When business loads increase, Auto Scaling automatically adds ECS instances to ensure sufficient computing capabilities. When business loads decrease, Auto Scaling automatically removes ECS instances to save costs. Auto Scaling also replaces unhealthy ECS instances to ensure service performance and business availability.

Additionally, Auto Scaling is seamlessly integrated with Server Load Balancer (SLB) and ApsaraDB RDS (RDS). This allows Auto Scaling to add or remove ECS instances to or from the backend server groups of the associated SLB instances, as well as to add or remove IP addresses of ECS instances to or from the whitelists of the associated RDS instances. Auto Scaling adapts to various complex scenarios without the need for manual operation and automatically processes business loads based on actual requirements. For more information, see Diagram of Auto Scaling.

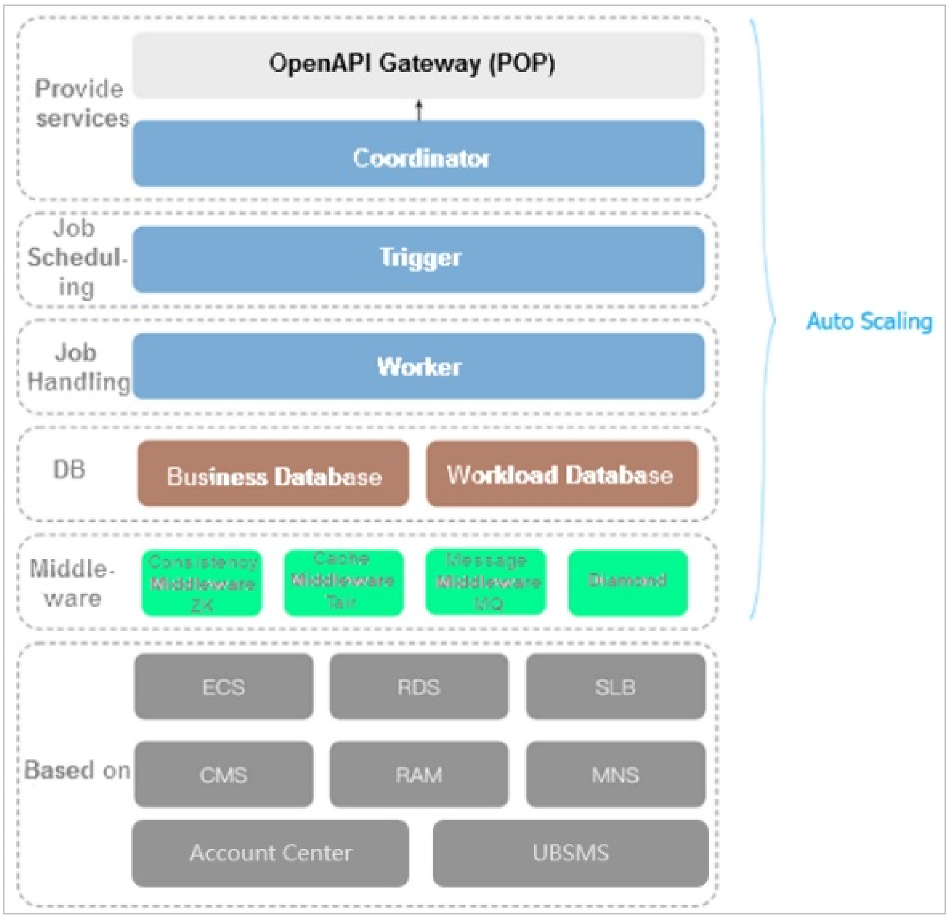
Diagram of Auto Scaling


1. Monitor health status
Scaling trigger task
Schedule task
Custom task
Health check task
Cloud Monitor task
Server Load Balancer
1. Monitor performance
2. ExecuteScalingRule (Rule ARI)
Scaling group
ECS
ECS
ECS
ECS
ECS
Scaling activity execution program
3.3 Create scaling activity
3.1 Compute the number of ECS instances to be added
3.2 Query the scaling configuration
5. Start the cool-down function for the scaling group
RDS
RDS
RDS
Scaling rules
Scaling configuration
Scaling activity
4.1 Create ECS instances
ECS
ECS
4.2 Configure Server Load Balancer and RDS

4.2. Architecture

Auto Scaling is a system that orchestrates ECS instances and provides services based on basic components such ECS. The Auto Scaling system consists of trigger, worker, database, and middleware services.

Diagram of the Auto Scaling architecture



Architecture description

Layer	Description
Middleware layer	ZooKeeper: ensures consistency by implementing distributed locks for Server Controller.
	Tair: provides caching services for Server Controller.
	Message Queue (MQ): provides message queuing services of VM statuses.
	Diamond: manages persistent configurations.
Database layer, which contains the business and workload databases	Worker: serves as the core of Auto Scaling. Auto Scaling receives a task and processes the task, including splitting the task, executing the task, and returning the execution results.
	Trigger: obtains information from health checks of instances and scaling groups, scheduled tasks, and Cloud Monitor to perform tasks scheduling.

Layer	Description
Public-facing services	Coordinator: serves as the ingress of the Auto Scaling architecture. It provides external management and control for services, processes API calls, and triggers tasks.
	Open API Gateway: provides basic services such as authentication and parameter passthrough.

4.3. Features

4.3.1. Scenarios

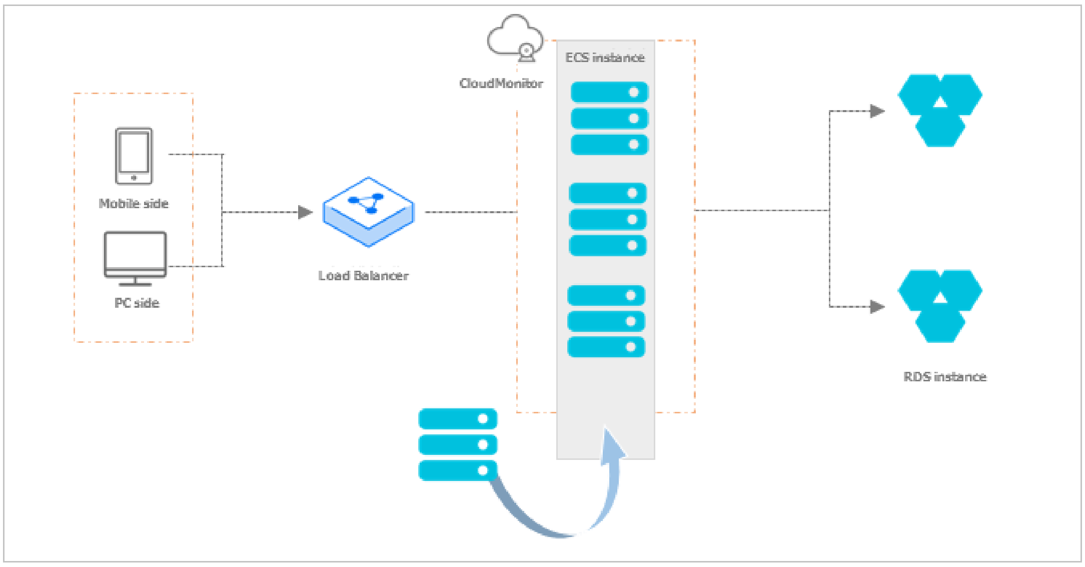
4.3.1.1. Overview

ESS automatically adjusts the number of elastic computing resources to meet fluctuating business demands. When business loads increase, ESS automatically adds ECS instances based on user-defined scaling rules to ensure sufficient computing capabilities. When business loads decrease, ESS automatically removes ECS instances to save costs.

4.3.1.2. Scale-out

When business loads surge above normal loads, Auto Scaling automatically increases underlying resources. This helps maintain the access speed and ensure that resources are not overloaded.

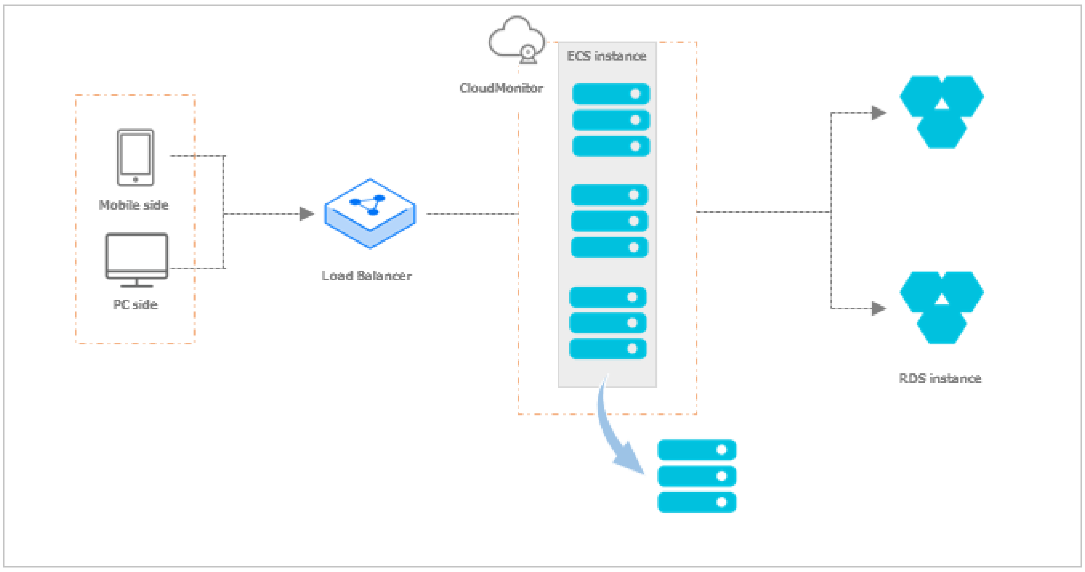
You can create scheduled tasks to perform automatic scale-out at specified points in time or configure Cloud Monitor to monitor ECS instance usage in real time and perform scale-out based on actual requirements. For example, when Cloud Monitor detects that the vCPU utilization of ECS instances in a scaling group exceeds 80%, Auto Scaling automatically scales out ECS resources based on user-defined scaling rules. During the scale-out event, Auto Scaling automatically creates ECS instances and adds these ECS 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.



4.3.1.3. Scale-in

When business loads decrease, Auto Scaling automatically releases underlying resources to prevent resource wastage and reduce costs.

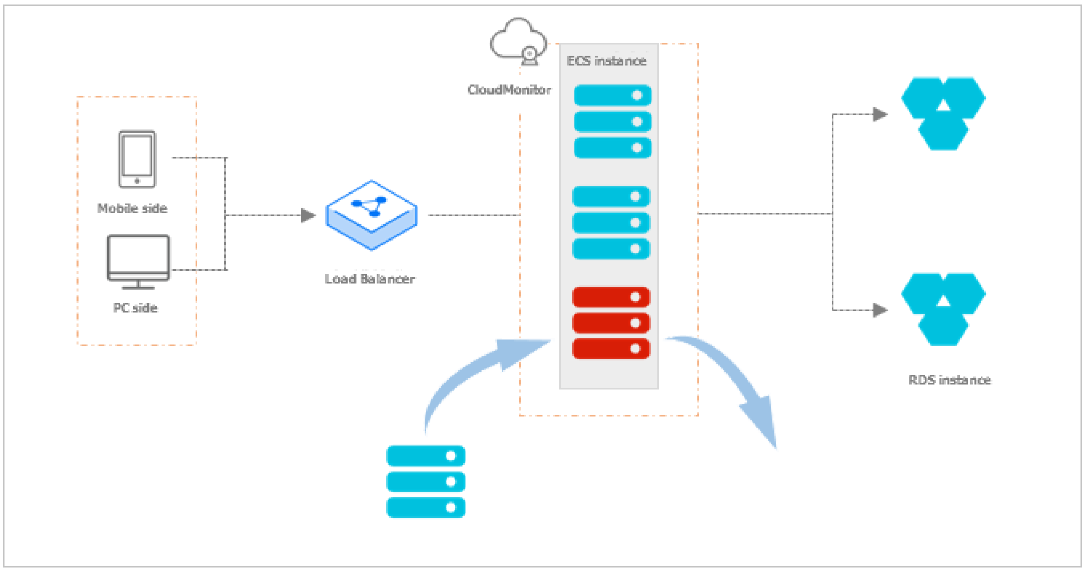
You can create scheduled tasks to automatically scale in ECS resources at specified points in time. You can also configure Cloud Monitor to monitor ECS instance usage in real time and scale in resources based on actual requirements. For example, when Cloud Monitor detects that the vCPU utilization of ECS instances in a scaling group is less than 30%, Auto Scaling automatically scales in ECS resources based on the scaling rule that you specified. During the scale-in event, Auto Scaling releases ECS instances and removes these ECS instances from 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-in event.



4.3.1.4. Elastic recovery

Auto Scaling provides the health check feature and automatically monitors the health status of ECS instances in a scaling group, so that the number of healthy ECS instances in the scaling group does not fall below the user-defined minimum value.

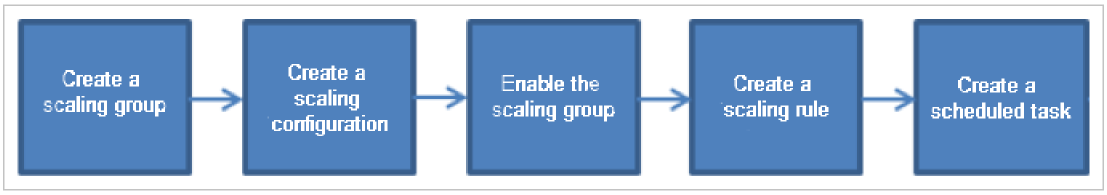
When Auto Scaling detects that an ECS instance is unhealthy, it automatically releases the unhealthy ECS instance, creates a new ECS instance, and adds the new instance to the backend server group of the associated SLB instance and the whitelist of the associated ApsaraDB RDS instance. The following figure shows the implementation of elastic recovery.



4.3.2. Components

To create a complete automatic scaling solution, you must create scaling groups, configurations, rules, as well as scheduled tasks or event-triggered tasks.

The following figure shows the procedure to create a complete automatic scaling solution.



Scaling groups

A scaling group is a group of ECS instances that are dynamically scaled based on the configured scenario. You can specify the minimum and maximum numbers of ECS instances in a scaling group, as well as the SLB and ApsaraDB RDS instances associated with the scaling group.

Scaling configurations

A scaling configuration is a template in Auto Scaling for creating ECS instances. When you create a scaling configuration, you must configure the parameters for creating an ECS instance, such as the instance type, image type, storage size, and Secure Shell (SSH) key pair that is used to log on to the ECS instance. You can also modify an existing scaling configuration.

Scaling rules

A scaling rule specifies a specific scaling activity, such as adding or removing ECS instances. The following scaling rules are supported:

- Change to N instances: After a scaling rule is executed, the number of ECS instances in a scaling group is adjusted to a specific value.

- Add N instances: After a scaling rule is executed, a specific number of ECS instances are added to the scaling group.
- Remove N instances: After a scaling rule is executed, a specific number of ECS instances are removed from the scaling group.

Scheduled tasks

A scheduled task specifies execution actions within a scaling group. It can trigger a specific scaling rule at a specific point in time to execute a scaling activity, such as adjusting the number of ECS instances in a scaling group.

Event-triggered tasks

Event-triggered tasks are scaling tasks associated with Cloud Monitor metrics and can be executed for automatic scaling in response to emergent or unpredictable business changes. After an event-triggered task is created and enabled, Auto Scaling collects monitoring data for the specified metric in real time and triggers an alert when the metric value meets the alert condition. Then, Auto Scaling executes the corresponding scaling rule to dynamically adjust the number of ECS instances in the scaling group.

5.Resource Orchestration Service (ROS)

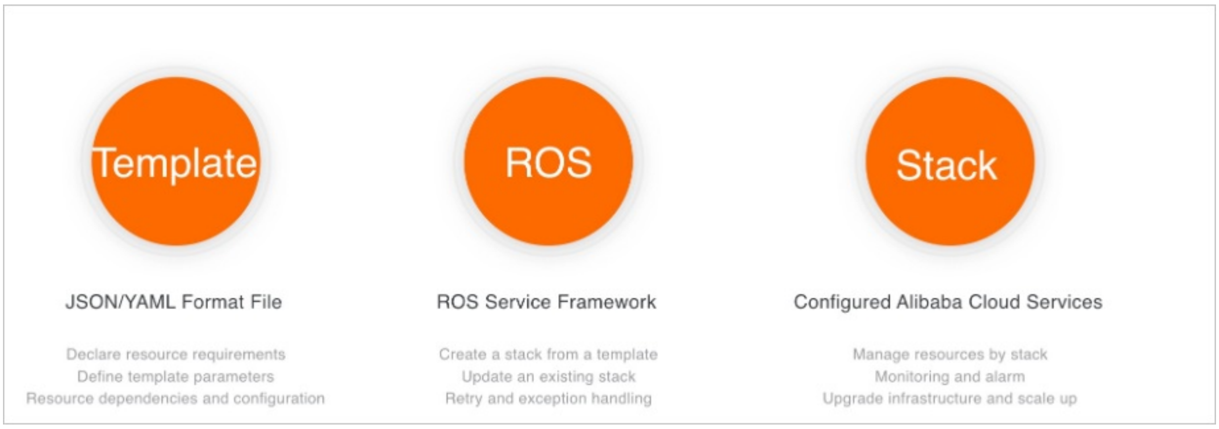
5.1. What is ROS?

Resource Orchestration Service (ROS) is an Apsara Stack service that can 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 Elastic Compute Service (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 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 or solution that encompasses cloud resources and applications. ISVs can use this method to integrate Apsara Stack 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 Apsara Stack resources. You can create, delete, and clone cloud resources by stack.


Template
ROS
Stack
JSON/YAML Format File
Declare resource requirements
Define template parameters
Resource dependencies and configuration
ROS Service Framework
Create a stack from a template
Update an existing stack
Retry and exception handling
Configured Alibaba Cloud Services
Manage resources by stack
Monitoring and alarm
Upgrade infrastructure and scale up

5.2. Benefits

This topic describes the benefits of Resource Orchestration Service (ROS). ROS provides a simple and convenient way to automate resource management.

You can use ROS to model and configure your cloud resources. After you create a template that defines your required resources such as Elastic Compute Service (ECS) and ApsaraDB RDS instances, ROS creates and configures these resources based on the template.

ROS provides flexible and convenient services at low costs. This allows you to focus on your core business and implement Infrastructure as Code (IaC). In DevOps practices, you can clone the development, test, and production environments to simplify the overall migration and scaling of applications.

ROS has the following benefits:

Automated resource orchestration

ROS creates and manages the lifecycle of cloud computing resources based on the defined templates of the cloud resources and their dependencies. It automates resource configuration and deployment, streamlines versioning, tracks resource changes, and simplifies cloud application delivery. ROS can be integrated with APIs and SDKs to provide automated O&M capabilities.

Simplified resource management

If you want to create a scalable web application that contains backend databases or to create a cluster that consists of dozens of ECS instances, you need to deploy multiple resources such as ECS, ApsaraDB RDS, Virtual Private Cloud (VPC), Auto Scaling (ESS), and Server Load Balancer (SLB) resources. Typically, you must deploy each resource and manually orchestrate the resources to satisfy your needs. These tasks are time-intensive and add complexity to the operations.

ROS allows you to create and manage your stacks and resources by using the following methods:

- Create or modify a template and define the resources and their dependencies in the template. ROS parses this template, creates the resources based on their dependencies and parameters, and automatically orchestrates the resources to satisfy your needs. This ensures that all resources created by using the template run properly.
- Adjust the stack template to fit your business needs.
- Delete all resources that are created by the same template in one click.
- Perform health checks on stacks in one click.

Quick replication of a collection of resources

If you have created a web application or cluster by using ROS, you can reuse the template to replicate the entire collection of resources. The template records the attributes and dependencies of each resource. You do not need to configure the resources again during resource replication.

Flexible integration with cloud products and services

You can use ROS to deploy and configure interactions between multiple cloud services. You can tailor your template based on your business and automated O&M requirements. ROS supports the following cloud products and services: ECS, ApsaraDB RDS, ApsaraDB for Memcache, KVStore for Redis, ApsaraDB for MongoDB, SLB, Object Storage Service (OSS), Log Service, Resource Access Management (RAM) and VPC.

Simple and visual creation of templates and stacks

ROS provides sample templates in the console to facilitate your operations. You can create a stack based on a sample template. During the stack creation, you need to configure only several parameters.

You can use Visual Editor to create and edit templates, and view the stack structure.

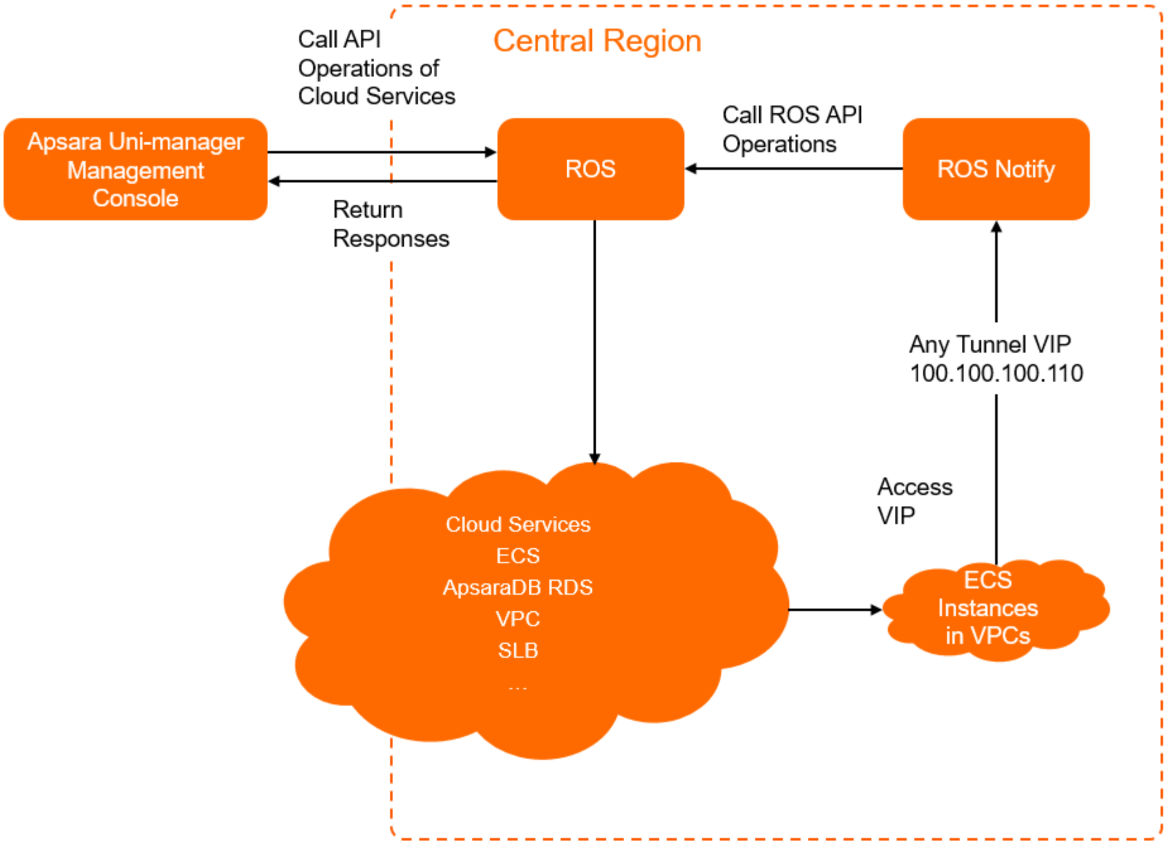
5.3. Architecture

This topic introduces the architecture of Resource Orchestration Service (ROS). You can use ROS by means of the Elastic Compute Service (ECS) console, API operations, and SDKs.

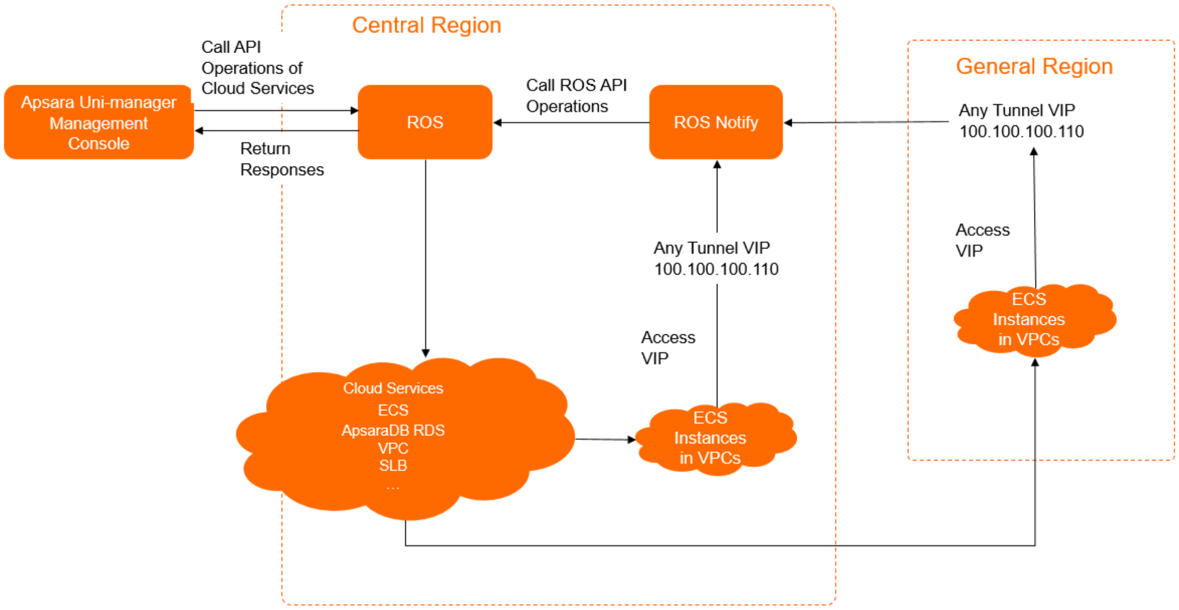
ROS supports the following cloud services: ECS, ApsaraDB RDS, ApsaraDB for MongoDB, ApsaraDB for Redis, ApsaraDB for Memcache, Server Load Balancer (SLB), Object Storage Service (OSS), Virtual Private Cloud (VPC), Elastic IP Address (EIP), Auto Scaling (ESS), Log Service, and Resource Access Management (RAM).

ROS supports single-region and multi-region deployment. The following figure shows the ROS architecture.

- Single-region deployment



- Multi-region deployment



5.4. Features

This topic describes the features of Resource Orchestration Service (ROS). The ROS engine automatically creates and configures all resources in a stack based on a template, which makes automatic deployment and Operations and Maintenance (O&M) possible.

ROS is an important service in cloud computing. You can define your cloud infrastructure as ROS templates and deploy them to the cloud from anywhere at any time to implement Infrastructure as Code (IaC). Compared with API calls of services, ROS allows you to process business resources in a more efficient way. Resources are created in stacks in ROS. You can also access these resources in separate consoles. You can manage stacks and their resources in the ROS console.

Create a stack

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 update cloud resources by using stacks. In DevOps scenarios, you can clone the development, test, and production environments. This simplifies the overall migration and scaling of applications.

Update a stack

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. You can modify the template and parameter configurations of the stack by updating the stack.

Recreate a 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.

Delete a stack

You can delete a stack that is no longer needed. You can choose whether to retain or release resources when you delete a stack.

6.Object Storage Service (OSS)

6.1. Introduction

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. Terms

This topic describes several basic terms used in OSS.

Object

The basic unit for data operations in OSS. Objects are also known as OSS files. An object is composed of object metadata, object content, and a key. A key can uniquely identify an object in a bucket. Object metadata is a group of key-value pairs that define the properties of an object, such as the last modification time and the object size. You can also assign user metadata to the object.

The lifecycle of an object starts when the object is uploaded, and ends when it is deleted. During the lifecycle, the object cannot be modified. OSS does not support modifying objects. If you want to modify an object, you must upload a new object with the same name as the existing object to replace it.

Note Unless otherwise stated, objects and files mentioned in OSS documents are collectively called objects.

Bucket

A container for OSS objects. Each object in OSS is contained in a bucket. You can configure and modify the attributes of a bucket to manage ACLs and lifecycle rules of the bucket. These attributes apply to all objects in the bucket. Therefore, you can create different buckets to meet different management requirements.

- OSS does not use a hierarchical structure for objects, but instead uses a flat structure. All elements are stored as objects in buckets. However, OSS supports folders as a concept to group objects and simplify management.
- You can create multiple buckets.
- A bucket name must be globally unique within OSS. Bucket names cannot be changed after the buckets are created.
- A bucket can contain an unlimited number of objects.

Strong consistency

A feature requires that object operations in OSS be atomic, which indicates that operations can only either succeed or fail. There are no intermediate states. To ensure that users can access only complete data, OSS does not return corrupted or partial data.

Object-related operations in OSS are highly consistent. For example, when a user receives an upload (PUT) success response, the uploaded object can be read immediately, and copies of the object have been written to multiple devices for redundancy. Therefore, there are no situations where data is not obtained when you perform the read-after-write operation. The same is true for delete operations. After you delete an object, the object and its copies no longer exist.

Similar to traditional storage devices, modifications are immediately visible in OSS while consistency is guaranteed.

Comparison between OSS and file systems

OSS is a distributed object storage service that stores objects based on key-value pairs. You can retrieve object content based on unique object keys. For example, object name *test1/test.jpg* does not necessarily indicate that the object is stored in a directory named test1. In OSS, *test1/test.jpg* is only a string. There is nothing essentially different between test1/test.jpg and *a.jpg*. Therefore, similar amounts of resources are consumed regardless of which object you access.

A file system uses a typical tree index structure. To access a file named *test1/test.jpg*, you must first access the test1 directory and then search for the *test.jpg* file in this directory. This makes it easy for a file system to support folder operations, such as renaming, deleting, and moving directories because these operations are only performed on directories. However, the performance of a file system depends on the capacity of a single device. The more files and directories that are created in the file system, the more resources and time are consumed.

You can simulate similar folder functions of a file system in OSS, but such operations are costly. For example, if you want to rename the test1 directory as test2, OSS must copy all objects whose names start with test1/ to generate objects whose names start with test2. This operation consumes a large amount of resources. Therefore, we recommend that you do not perform such operations in OSS.

Objects stored in OSS cannot be modified. A specific API operation must be called to append an object, and the generated object is different from objects uploaded by using other methods. To modify even a single byte, you must upload the entire object again. A file system allows you to modify files. You can modify the content at a specified offset location or truncate the end of a file. These features make file systems suitable for more general scenarios. However, OSS supports a large amount of concurrent access, whereas the performance of a file system is subject to the performance of a single device.

We recommend that you do not map operations on OSS objects to file systems because it is inefficient. If you attach OSS as a file system, we recommend that you only add new files, delete files, and read files. You can make full use of OSS advantages, such as the capability to process and store large amounts of unstructured data such as images, videos, and documents.

6.1.3. Benefits

OSS provides secure, cost-effective, and high-durability services for you to store large amounts of data in the cloud. This topic compares OSS with the traditional user-created server storage to help you better understand the benefits of OSS.

Advantages of OSS over self-managed server storage

Item	OSS	Self-managed server storage
Reliability	Supports automatic backup for data redundancy.	• Prone to errors due to low hardware reliability. If a disk has a bad sector, data may be lost. • Manual data restoration is complex and requires a lot of time and technical resources.
Security	• Provides multi-level security protection for enterprises. • Provides resource isolation mechanisms for multiple tenants and supports geo-disaster recovery. • Provides various authentication and authorization mechanisms. It also provides features such as whitelists, hotlink protection, RAM, and Security Token Service (STS) for temporary access.	• Requires additional scrubbing devices and black hole policy-related services. • Requires a separate security mechanism.
Data processing	Provides Image Processing (IMG).	Data processing capabilities must be purchased and separately deployed.

More benefits of OSS

- Secure

  OSS features a comprehensive permission control mechanism and provides multiple encryption algorithms and methods. OSS complies with the regulations of multiple organizations including the U.S. Securities and Exchange Commission (SEC) and Financial Industry Regulatory Authority, Inc. (FINRA), delivering services that can meet the requirements of your enterprise on data security and compliance.

- Reliable

  OSS adopts redundant data storage mechanisms and supports multiple features such as cross-region replication and cross-cloud replication to provide highly reliable storage service based on objects.

- Stable

Apsara Stack OSS is a stable, secure, and reliable storage service that is built based on Apsara Infrastructure Management Framework and Apsara Distributed File System. OSS is the core infrastructure of data storage for Alibaba Group. This reliable and highly available service is one of the backbone services that ensure stability during the peak hours of Double 11 shopping festival. OSS features a multi-redundant architecture to provide reliable data storage. In addition, OSS is built on a high-availability architecture to eliminate single points of failure (SPOFs) and ensure the continuity of your services.

- Intelligent

  OSS allows you to configure lifecycle rules to intelligently manage data in its entire lifecycle. OSS also supports multiple data processing features, including Image Processing (IMG), video snapshot, and document preview. You can use this features to meet manage and analyze the data of your enterprise and reduce development costs.

6.1.4. Scenarios

This topic describes the application scenarios of OSS.

Massive storage for image, audio, and video applications

OSS can be used to store large amounts of data, such as images, audio and video data, and logs. Various devices, websites, and mobile applications can directly read data from or write data to OSS. You can write data to OSS by uploading files or using streams.

Dynamic and static content separation for websites and mobile applications

By using the BGP bandwidth, you can download data with an ultra-low latency.

Offline data storage

OSS provides storage with low cost and high availability. Therefore, you can use OSS to store enterprise data that needs to be archived offline for a long period.

Cross-region disaster recovery

You can use cross-region replication (CRR) or cross-cloud replication to asynchronously replicate your data between two clusters or clouds in near real time. This way, you can build a storage architecture with three data centers in two regions to store your data in different regions for backup and disaster recovery, which ensures the continuity of your business when extreme disaster events occur.

6.1.5. Features

OSS provides secure, cost-effective, and high-durability services for you to store large amounts of data in the cloud. This topic describes the features supported by OSS.

Category	Feature	Description
	Create buckets	Before you upload an object to OSS, you must create a bucket to store the object.
	Delete buckets	If you no longer use a bucket, delete it to avoid further fees.

Category	Feature	Description
Bucket management	Configure bucket 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.
	Configure static website hosting	You can use the static website hosting feature to host your static website on an OSS bucket and use the endpoint of the bucket to access the website.
	Configure hotlink protection	To prevent additional fees caused by unauthorized access to the data in your bucket, you can hotlink protection for your buckets based on the Referer field in HTTP requests.
	Configure logging	When you access OSS, large numbers of access logs are generated. After you configure logging for a bucket, OSS generates log objects every hour in accordance with a predefined naming convention and then stores the access logs as objects in a specified bucket.
	Configure CORS	OSS provides cross-origin resource sharing (CORS) over HTML5 to implement cross-origin access.
	Configure data encryption	OSS allows you to encrypt uploaded data on the OSS server.
	Configure lifecycle rules	You can create and manage lifecycle rules for all or a subset of objects in a bucket. You can configure lifecycle rules to manage multiple objects and automatically delete parts.
Object management	Upload objects	You can upload all types of objects to a bucket.
	Create directories	You can manage OSS directories the way you manage directories in Windows.
	Search for objects	You can search for objects whose names contain the same prefix in a bucket or directory.
	Obtain object URLs	You can obtain the URL of an object to share or download the object.
	Configure object tagging	OSS allows you to configure object tagging to classify 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.
	Delete objects	You can delete a single object or multiple objects.
	Delete directories	You can delete directories.
	Manage parts	You can delete all or some parts from a bucket.

Category	Feature	Description
Data processing	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 a valid origin for the bucket, you can obtain the data based on the rules.
	Process images	You can add Image Processing (IMG) parameters to GetObject requests to process image objects stored in OSS. For example, you can add image watermarks to images or convert image formats.
Access control	Configure object ACLs	You can configure the access control list (ACL) of an object when you upload the object or modify the ACL of an uploaded object.
	Configure bucket ACLs	OSS supports ACL for access control. You can configure the ACL of a bucket when you create the bucket or modify the ACL of a created bucket.
	Use STS to authorize temporary access	You can use Security Token Service (STS) to grant a third-party application or a Resource Access Management (RAM) user an access credential with a custom validity period and permissions.
	Add the Authorization header to sign a request	You can include the Authorization header in an HTTP request to carry signature information and indicate that the requester has been authorized.
	Add a signature to a URL	You can add signature information to a URL and provide the URL to a third-party user for authorized access.
Data encryption	Configure server-side encryption	OSS allows you to encrypt uploaded data on the OSS server.
	Configure client-side encryption	You can encrypt objects on local clients before you upload the objects to OSS.
Disaster recovery	Configure zone-disaster recovery	After you enable zone-disaster recovery for a primary bucket, a secondary bucket whose name is the same as that of the primary bucket is automatically created. Information stored in the primary bucket is automatically synchronized to the secondary bucket.
	Configure cross-cloud replication	You can use the cross-cloud replication feature to synchronize OSS data between two clouds.
	Configure CRR	You can configure cross-region replication (CRR) to synchronize operations such as the creation, overwriting, and deletion of objects from the source bucket to the destination bucket.

Category	Feature	Description
Data monitoring	Monitor basic data	You can monitor the basic data metrics of a bucket, including the used traffic, number of requests, 5XX requests, and service level agreement (SLA).

6.2. Security and compliance

6.2.1. Access control

6.2.1.1. Configure hotlink protection

You can configure a Referer whitelist for a bucket to prevent your resources in the bucket from unauthorized access.

The hotlink protection function allows you to configure a Referer whitelist for a bucket and select whether to allow empty Referer field. This way, only requests from the domain names that are included in the Referer whitelist can access the data in the bucket. OSS allows you to configure Referer whitelists based on the Referer header field in HTTP and HTTPS requests.

The following scenarios describe whether to use hotlink protection to verify access to OSS:

- Only anonymous requests and requests that contains signed URLs are verified.
- Requests that contain the `Authorization` header field are not verified.

OSS determines the source from which a request is sent based on the Referer header field in the request. When a browser sends a request to the web server, the Referer field is contained in the request to indicate the source from which the request is sent. OSS determines whether to allow or deny a request based on the Referer field contained in the request and the Referer whitelist configured for the specified bucket. If the Referer field in the request matches the Referer whitelist, the request is allowed. Otherwise, the request is denied. Example: The Referer whitelist configured for a bucket includes only *https://10.10.10.10.com*.

- User A adds an image object named test.jpg to website *https://10.10.10.10.com*. When a user accesses the image on the website, the browser sends a request in which the value of the Referer field is *https://10.10.10.10.com*. OSS allows the request because the Referer field in the request is included in the Referer whitelist.
- User B adds the URL of the image object to the website *https://127.0.0.1.com* without authorization. When a user accesses the image on the website, the browser sends a request in which the value of Referer field is *https://127.0.0.1.com*. OSS denies the request because the Referer field in the request is excluded in the Referer whitelist.

6.2.1.2. RAM Policy

Resource Access Management (RAM) policies are configured based on users. You can configure RAM policies to control the resources that can be accessed by users.

For example, you can configure the following RAM policy to grant a RAM user permissions to only read objects in the `myphotos/hangzhou/2014/` and `myphotos/hangzhou/2015/` directories:

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "oss:GetObject"
      ],
      "Resource": [
        "acs:oss:*:*:myphotos/hangzhou/2014/*",
        "acs:oss:*:*:myphotos/hangzhou/2015/*"
      ]
    }
  ]
}
```

6.2.1.3. CORS

Cross-origin resource sharing (CORS) is a standard cross-origin solution provided by HTML5 to allow web application servers to control cross-origin access, which ensures the security of data transmission across origins.

Browsers check cross-origin requests based on the same-origin policy to keep the website content secure. When a request is sent from Website A by using JavaScript to access Website B of another origin, the browser rejects the request. In this case, you can configure CORS rules to allow cross-origin requests.

For example, two different origins run on the same browser, such as www.example.com and www.test.com. The origins access the same resource from another origin. If the server first receives the request from www.example.com, the server includes the Access-Control-Allow-Origin header in the response to the corresponding user. When the request from www.test.com is sent, the browser returns the last cached response to the user. The header content does not match that of CORS rules. As a result, the request from www.test.com fails.

The same-origin policy is a key security mechanism that isolates potential malicious files. It prevents scripts and documents loaded from different origins from interacting with each other. Origins that use the same protocol, domain name or IP address, and port number are considered to be the same origin. The following table lists examples and checks whether the examples and *http://www.aliyun.com/org/test.html* are from the same origin.

URL	Access result	Cause
http://www.aliyun.com/org/other.html	Successful	Same protocol, domain name, and port number
http://www.aliyun.com/org/internal/page.html	Successful	Same protocol, domain name, and port
https://www.aliyun.com/page.html	Failed	Different protocols (HTTP and HTTPS)
http://www.aliyun.com:22/dir/page.html	Failed	Different port numbers (22 and 80)

URL	Access result	Cause
http://help.aliyun.com/dir/other.html	Failed	Different domain names

6.2.2. Data encryption

6.2.2.1. Server-side encryption

Object Storage Service (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.

Encryption methods

OSS protects static data by using server-side encryption. You can use this method in scenarios in which additional security or compliance is required, such as the storage of deep learning samples and online collaborative documents.

Only one server-side encryption method can be used for an object at a time. OSS provides the following server-side encryption methods that you can use in different scenarios:

- Server-side encryption by using Key Management Service (SSE-KMS)

  You can use the default customer master key (CMK) or specify a CMK to encrypt or decrypt large amounts of data. This method is cost-effective because you do not need to send user data to the KMS server over networks for encryption or decryption.

  > Notice
  >
  > - The key used to encrypt the object is also encrypted and written into the metadata of the object.
  > - Server-side encryption that uses the default CMK (SSE-KMS) only encrypts the data in the object. The metadata of the object is not encrypted.

- Server-side encryption by using OSS-managed keys (SSE-OSS)

  You can use SSE-OSS to encrypt each object. To improve security, OSS uses master keys to encrypt data keys that are rotated on a regular basis. You can use this method to encrypt and decrypt multiple objects at a time.

6.2.2.2. Client-side encryption

If client-side encryption is performed, objects are encrypted on the local client before they are uploaded to OSS. This topic describes how to perform client-side encryption.

In client-side encryption, a random data key is generated for each object to perform symmetric encryption on the object. The client uses a CMK to encrypt the random data key. The encrypted data key is uploaded as a part of the object metadata and stored in the OSS server. When an encrypted object is downloaded, the client uses the CMK to decrypt the random data key and then uses the data key to decrypt the object. The CMK is used only on the client and is not transmitted over the network or stored in the server, which ensures data security.

> Note
>
> - Client-side encryption supports multipart upload for objects larger than 5 GB. When you use multipart upload to upload an object, you must specify the total size of the object and the size of each part. The size of each part except for the last part must be the same and be a multiple of 16 bytes.
> - After you upload objects encrypted on the client, object metadata related to client-side encryption is protected. In this case, CopyObject cannot be used to modify object metadata.

Disclaimer

- When you use client-side encryption, you must ensure the integrity and validity of the CMK. If the CMK is incorrectly used or lost due to improper maintenance, you will be held responsible for all losses and consequences caused by decryption failures.
- When you copy or migrate encrypted data, you must ensure the integrity and validity of the object metadata related to client-side encryption. If the encrypted metadata is incorrectly used or lost due to improper maintenance, you will be held responsible for all losses and consequences caused by decryption failures.

6.2.3. Log management

When you access OSS, large numbers of access logs are generated. After you enable and configure logging for a bucket, OSS generates log objects every hour in accordance with a predefined naming convention and then stores the access logs as objects in a specified bucket. You can use Apsara Stack Log Service or build a Spark cluster to analyze the logs.

6.3. Data reliability assurance

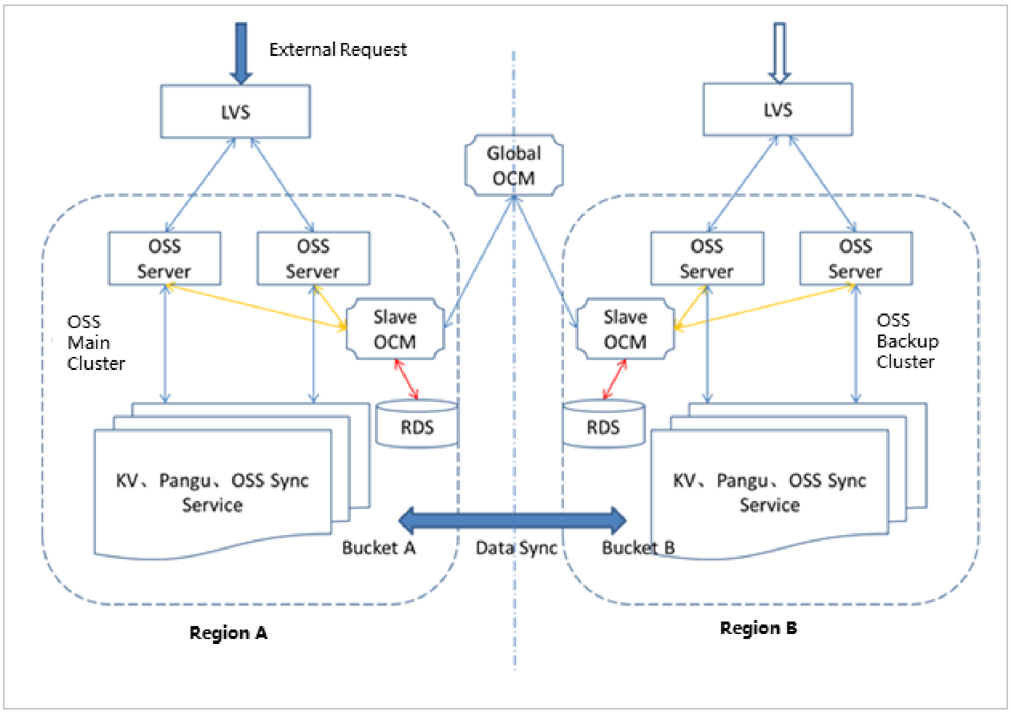
6.3.1. Disaster recovery and backup

6.3.1.1. CRR

Cross-region replication (CRR) provides automatic and asynchronous (near real-time) replication of objects across buckets in different Object Storage Service (OSS) regions. Operations such as creating, overwriting, and deleting objects can be synchronized from a source bucket to a destination bucket.

Objects in the destination bucket are exact replicas of those in the source bucket. They have the same object names, versioning information, object content, and object metadata such as the creation time, owner, user metadata, and object access control lists (ACLs). When you use CRR, you can query the replication progress in real time. This feature displays the last synchronization time for real-time data synchronization and the percentage of synchronization for historical data migration.

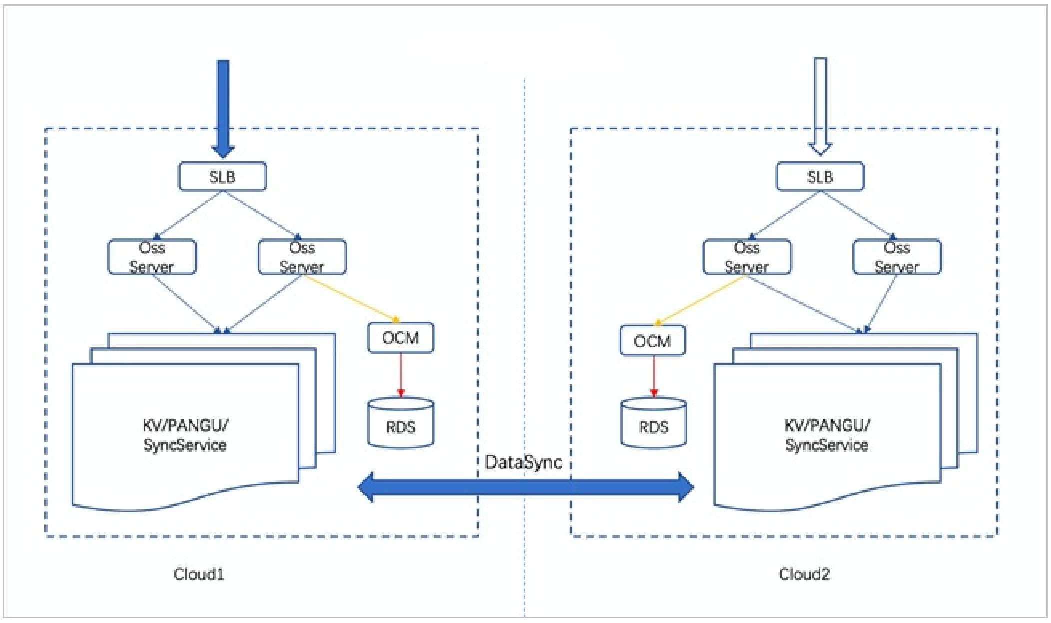
You can use CRR to build a backup solution in which data is backed up in a data center that is hundreds of kilometers away from your local data center. This way, you can meet the requirements on compliance and remote data backup and improve the continuity of your business. When a region becomes unavailable due to disaster events, you can switch over your business to the backup region. You can use CRR to reduce the cost of building a remote data backup center.



6.3.1.2. Cross-cloud replication

You can use cross-cloud replication to replicate data from a cloud to another cloud. This way, you can back up data in a cloud. When a cloud fails, you can switch over your business to another cloud to ensure business continuity.

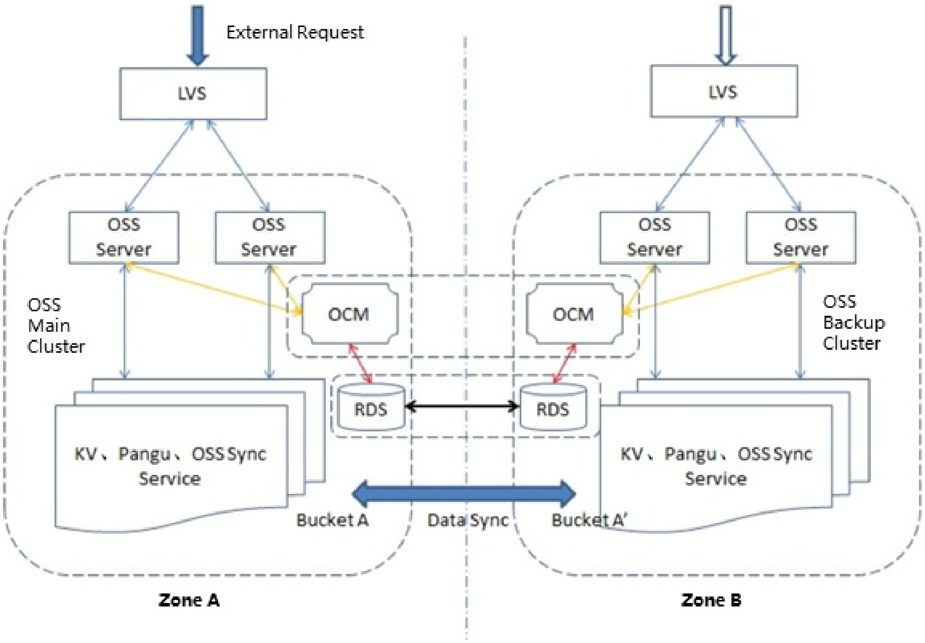
You can also use cross-cloud replication to migrate data between clouds. When the storage capacity of a cloud is insufficient, you can use cross-cloud replication to replicate data from a bucket in the cloud to a bucket in another cloud.



6.3.1.3. Zone-disaster recovery

Zone-disaster recovery allows you to store multiple replicas of your data in multiple zones of the same region. This feature protects your data from being lost and helps you recover your business when a single zone fails.

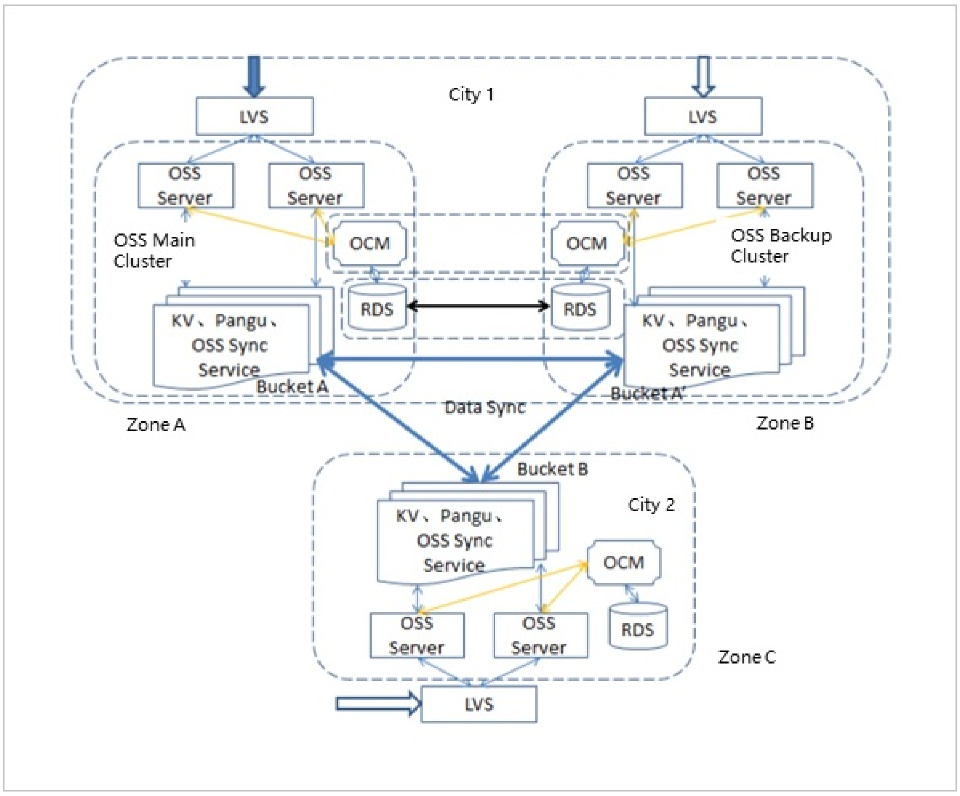
When you use zone-disaster recovery, you must create a primary zone in which your business runs and a secondary zone in the same region. Data that you write to buckets in the primary zone is asynchronously replicated to the secondary zone. When the primary zone becomes unavailable due to network disconnections, power outages, or other disaster events, you can switch over your business to the secondary zone with one click.



6.3.1.4. Three data centers across two regions

If your business has high requirements on data backup, you can use zone-disaster recovery and cross-region replication to build a disaster recovery solution based with three data centers across two regions.

In the solution, you can create a primary zone and a secondary zone in the local region and create a secondary zone in a remote region. When you write data to buckets in the primary zone, OSS asynchronously replicates the data to the secondary zones in the local region and remote region. When the secondary zone in the same region is unavailable because of accidents, you can switch over your business to the secondary zone in the remote region.



6.4. Data processing

6.4.1. Image processing

You can add Image Processing (IMG) parameters to GetObject requests to process image objects stored in Object Storage Service (OSS). For example, you can add image watermarks to images or convert image formats.

OSS allows you to directly use one or more parameters to process images. You can also encapsulate multiple IMG parameters in a style to batch process images. When multiple IMG parameters are specified, OSS processes the image in the order of the parameters.

You can use object URLs, API operations, and SDKs to process images. The following table describes the IMG operations supported by OSS.

IMG operation	Parameter	Description
Resize images	resize	Resizes images to a specified size.
Incircle	circle	Crops images based on the center point of images to ellipses of the specified size.
Custom crop	crop	Crops rectangular images of the specified size.

IMG operation	Parameter	Description
Indexed cut	indexcrop	Cuts images along the specified horizontal or vertical axis and selects one of the images.
Rounded rectangle	rounded-corners	Crops images to rounded rectangles based on the specified rounded corner size.
Automatic rotation	auto-orient	Auto-rotates images for which the auto-orient parameter is configured.
Rotate	rotate	Rotates images clockwise based on the specified angle.
Blur	blur	Blurs images.
Adjust brightness	bright	Adjusts the brightness of images.
Sharpen	sharpen	Sharpens images.
Adjust contrast	contrast	Adjusts the contrast of images.
Gradual display	interlace	Configures gradual display for the JPG images.
Adjust image quality	quality	Adjusts the quality of images in the JPG and WebP formats.
Convert format	format	Converts image formats.
Add watermarks	watermark	Adds image or text watermarks to images.
Query average tone	average-hue	Queries the average tone of images.
Query image information	info	Queries image information, including basic information and EXIF information.

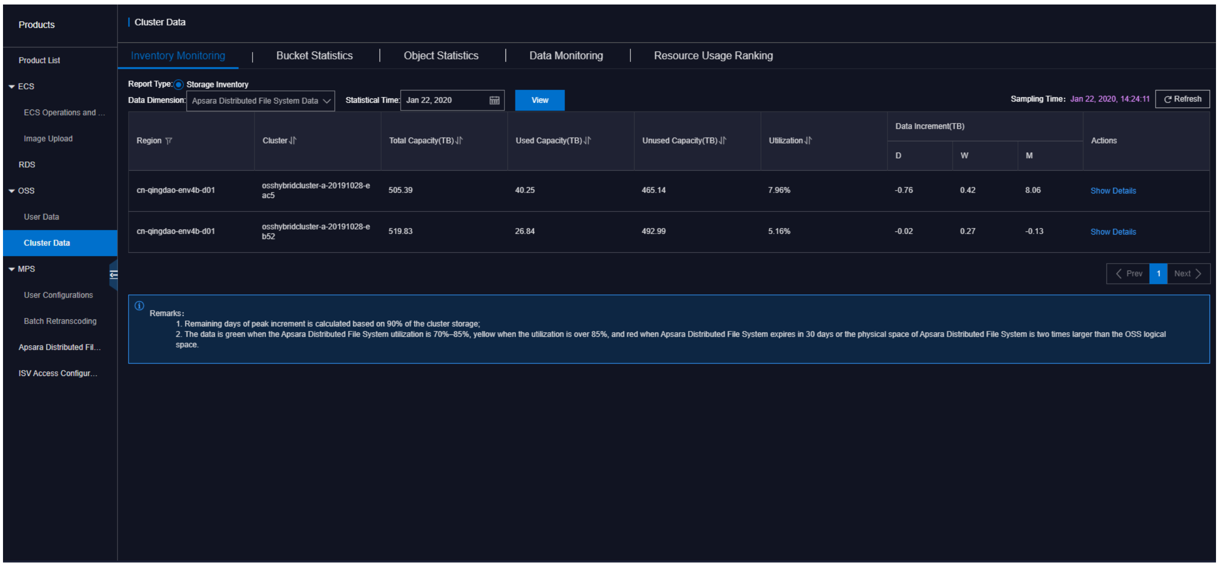
6.5. Basic data monitoring

6.5.1. O&M dashboard

6.5.1.1. Inventory monitoring

You can monitor the following metrics: total capacity, unused capacity, used capacity, and storage utilization.

The following figure shows the inventory monitoring page of an OSS cluster.



Report Type: ● Storage Inventory

Data Dimension: Apsara Distributed File System Data

Statistical Time: Jan 22, 2020

Sampling Time: Jan 22, 2020, 14:24:11

Region	Cluster	Total Capacity(TB)	Used Capacity(TB)	Unused Capacity(TB)	Utilization	Data Increment(TB)			Actions
						D	W	M	
cn-qingdao-env4b-d01	osshybridcluster-a-20191028-eac5	505.39	40.25	465.14	7.96%	-0.76	0.42	8.06	Show Details
cn-qingdao-env4b-d01	osshybridcluster-a-20191028-eb52	519.83	26.84	492.99	5.16%	-0.02	0.27	-0.13	Show Details

Remarks:

1. Remaining days of peak increment is calculated based on 90% of the cluster storage;
2. The data is green when the Apsara Distributed File System utilization is 70%–85%, yellow when the utilization is over 85%, and red when Apsara Distributed File System expires in 30 days or the physical space of Apsara Distributed File System is two times larger than the OSS logical space.

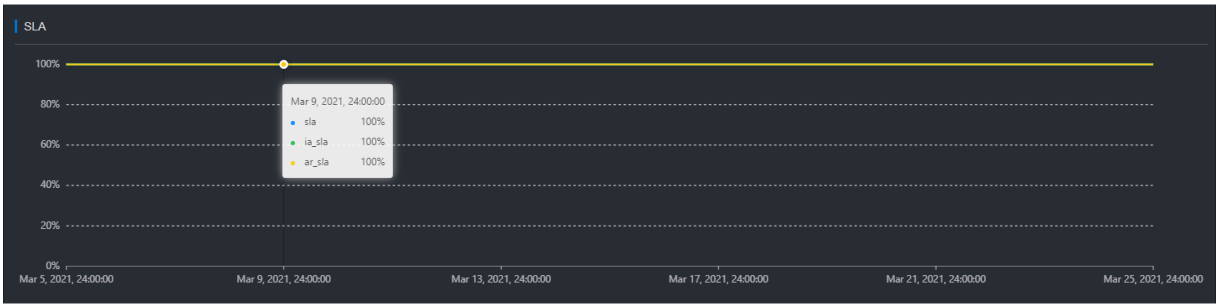
- **Total Capacity**: the total physical storage capacity of the OSS cluster.
- **Used Capacity**: the used physical storage capacity of the OSS cluster.
- **Unused Capacity**: the available physical storage capacity of the OSS cluster.
- **Usage**: the storage usage of the OSS cluster
- **Data Increment**: the growth rates of the storage usage of the cluster on a daily, weekly, and monthly basis. You can use this metric to estimate when to expand the storage of the cluster to accommodate your business.

6.5.1.2. Key OSS data monitoring

You can view the information about OSS on the O&M dashboard, including service level agreement (SLA), traffic, query per second (QPS), and latency.

SLA

The SLA of OSS is measured by the success rate of requests sent from clients to OSS servers. The SLA of OSS can be calculated based on the following formula: Number of non-5XX requests per 10 seconds or an hour/Number of valid requests x 100%. If the SLA of OSS does not reach 100%, contact the technical support for an in-depth check of your cluster.



Traffic

You can view the inbound and outbound traffic over the Internet, private networks, and Content Delivery Network (CDN) as well as the synchronization traffic on the dashboard. You can understand the load of a cluster based on these traffic metrics.

QPS

You can view the number of billed requests and the number of the following requests on the dashboard: CopyObject, GetObject, PutObject, UploadPart, PostObject, AppendObject, HeadObject, and GetObjectInfo. You can understand the load of a cluster based on these QPS metrics.

Latency

You can view the response latency of all requests sent from clients to OSS on the dashboard. Higher cluster loads lead to longer latency. The latency varies with the network connection conditions within or outside the cluster. If you experience high latency, contact the technical support for an in-depth check of OSS and your networks.

6.5.2. Alert dashboard

6.5.2.1. View alerts

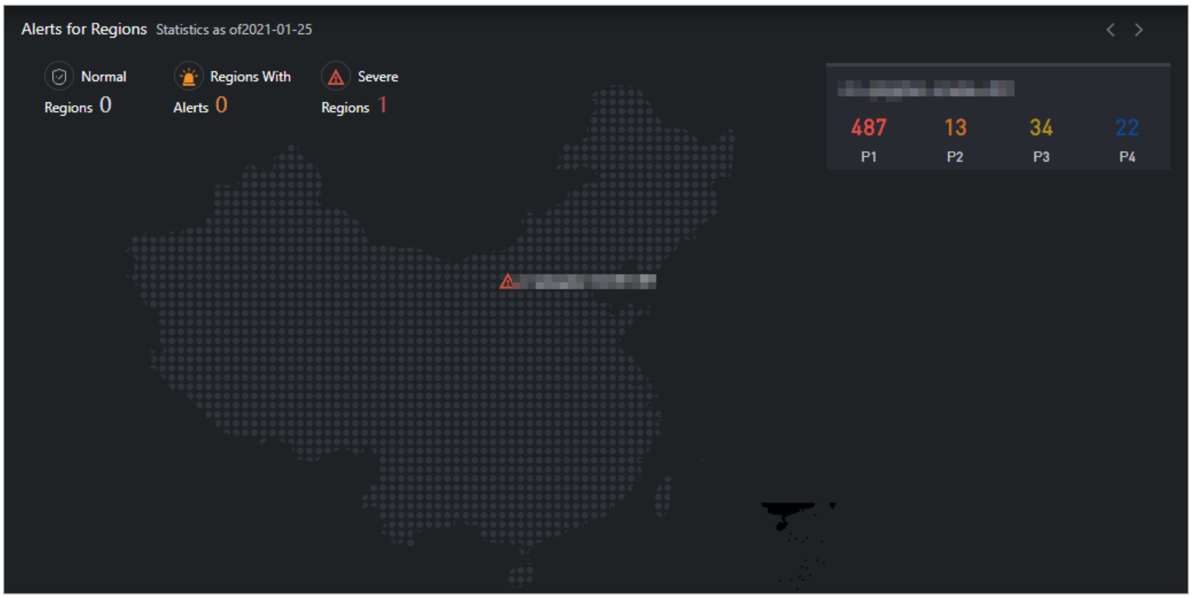
This topic describes how to view alerts within each region of a multi-region scenario. You can view the details of alerts related to OSS within a specific period on the alert dashboard.

Alert overview

You can view the statistics and distribution of alerts in different regions on the alert dashboard.

- In the **Alerts for Regions** section, view the distribution of alerts in different regions.

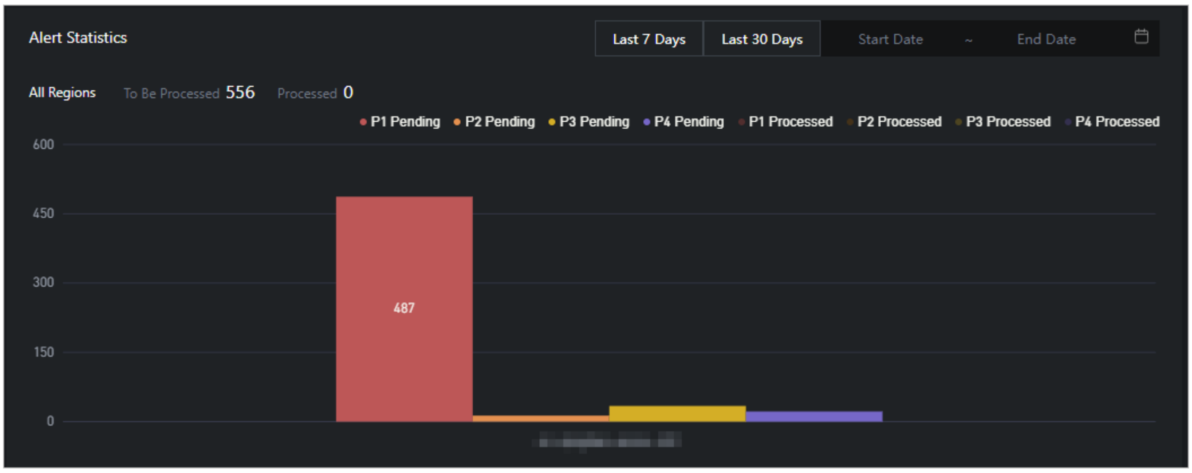
  Move the pointer over a region with alerts, and the specific number of alerts is displayed.



Note P1, P2, P3, and P4 have the following meanings:

- P1: urgent alerts
- P2: major alerts
- P3: minor alerts
- P4: reminder alerts

- In the **Alert Statistics** section, view the statistical data of alerts in the region.

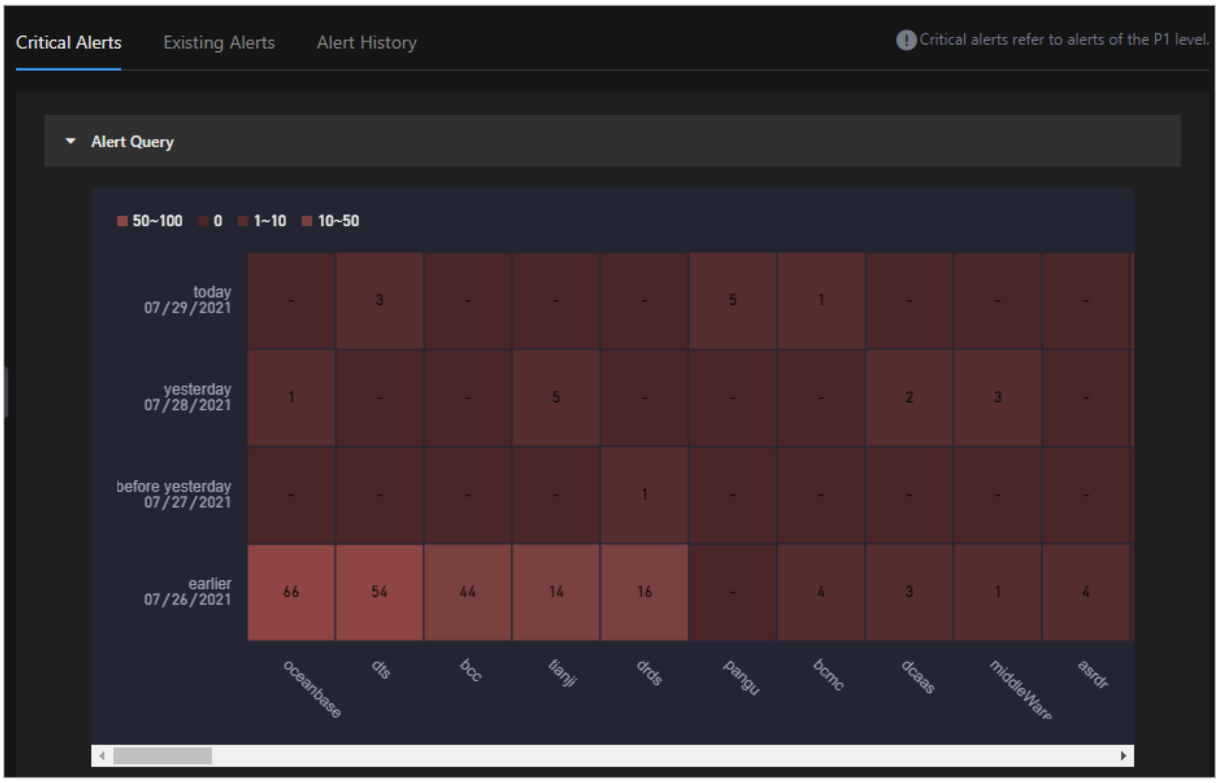


Alert details

On the **Alerts** page, you can view the detail information about alerts to locate and troubleshoot issues on site. If P1 or P2 alerts are not handled for a long period of time, the availability of the service may be affected.

Notice We recommend that you contact on-site engineers or technical support to handle alerts but not handle alerts by yourself.

- Click the **Critical Alerts**, **Existing Alerts**, and **Alert History** tabs to view the information about different types of alerts.
- The different colors of alerts indicate different ranges of quantities.
- The different colors of alerts indicate different ranges of quantities.



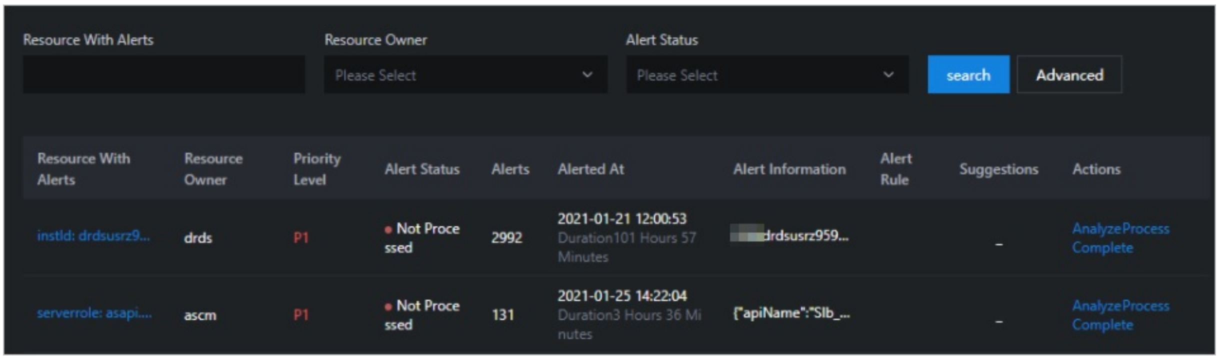
Critical alerts refer to alerts of the P1 level.

■ 50~100 ■ 0 ■ 1~10 ■ 10~50

	oceanbase	dts	bcc	tianji	drds	pangu	bcmc	dcaas	middleWare	asrdr
today 07/29/2021	-	3	-	-	-	5	1	-	-	-
yesterday 07/28/2021	1	-	-	5	-	-	-	2	3	-
before yesterday 07/27/2021	-	-	-	-	1	-	-	-	-	-
earlier 07/26/2021	66	54	44	14	16	-	4	3	1	4

Enter an alert resource ID in the **Resource With Alerts** search box, select the required options from the **Resource Owner** and **Alert Status** drop-down lists, and then click **Search** to view the alerts.

Move the pointer over an alert resource or a piece of alert information, and the full description of the alert information is displayed.



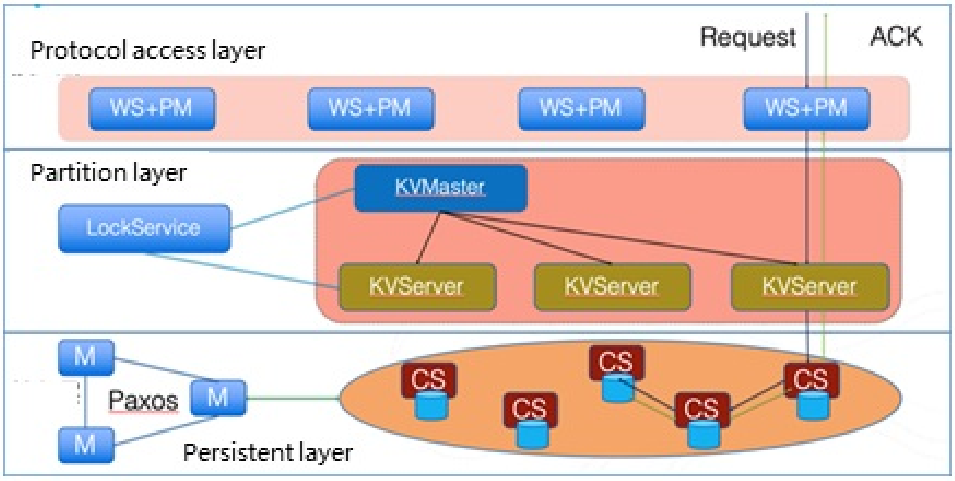
Resource With Alerts	Resource Owner	Priority Level	Alert Status	Alerts	Alerted At	Alert Information	Alert Rule	Suggestions	Actions
instId: drdsusrz9...	drds	P1	● Not Processed	2992	2021-01-21 12:00:53 Duration101 Hours 57 Minutes	[redacted]drdsusrz959...		-	AnalyzeProcess Complete
serverrole: asapi....	ascm	P1	● Not Processed	131	2021-01-25 14:22:04 Duration3 Hours 36 Minutes	{"apiName":"Slb_...		-	AnalyzeProcess Complete

6.6. Architecture

6.6.1. System architecture

OSS is a storage solution that is built on the Apsara system. It is based on the infrastructure such as Apsara Distributed File System and SchedulerX. This infrastructure provides OSS and other Alibaba Cloud services with important features such as distributed scheduling, high-speed networks, and distributed storage.

The following figure shows the system architecture of OSS.

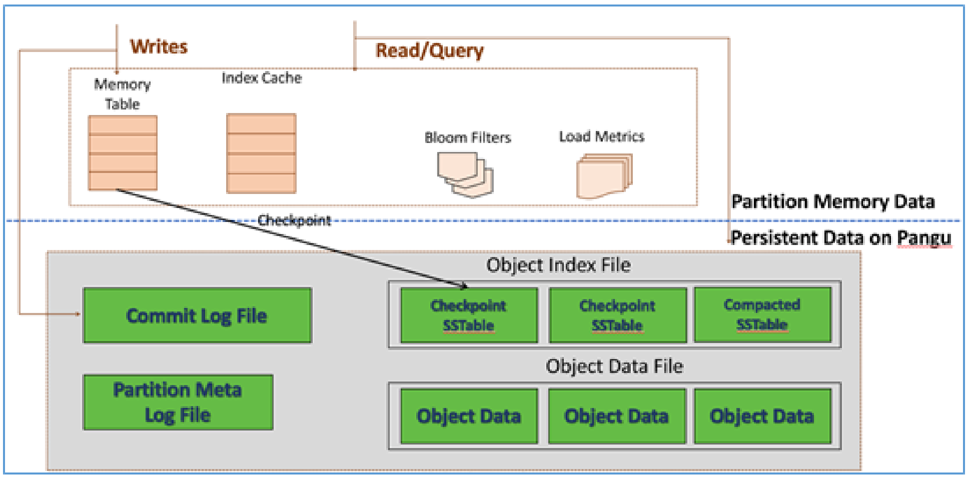


The OSS architecture is composed of three layers: protocol access layer, partition layer, and persistent storage layer.

- Protocol access layer
  - WS: uses the open-source Tengine component, and provides HTTP and HTTPS for external services.
  - PM: parses the HTTP request as the read/write operation on the back-end KV or another module. PM also receives and authenticates the user request sent through a RESTful protocol. If the authentication succeeds, the request is forwarded to KV Engine for further processing. If the request fails the authentication, an error message is returned.
- Partition layer

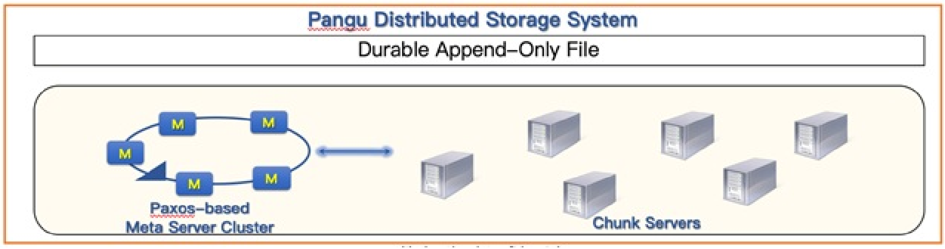
  The partition layer uses a highly scalable storage system with high performance and strong consistency to manage the index of a large amount of data. The index of objects stored in OSS is managed by range partitions. OSS divides the index of a large amount of objects into range partitions based on the loads and assigns partition servers to manage the range partitions. In addition, the partition layer supports a large number of concurrent requests and provides the automatic load balancing and garbage collection features.

  The index system uses the LSM tree structure that consists of KVMasters and KVServers. KVMaster manages and schedules partitions. KVServer stores indexes and actual data of partitions.



- Persistent layer

The persistent layer provides a Paxos-based distributed file system that can store exabytes of data and provide high reliability and availability. Masters in this layer use the Paxos protocol to ensure that the metadata stored on the masters is consistent. This way, objects can be effectively stored and accessed in a distributed manner. In addition, data stored in the file system is backed up for redundancy and can be recovered when software or hardware errors occur.

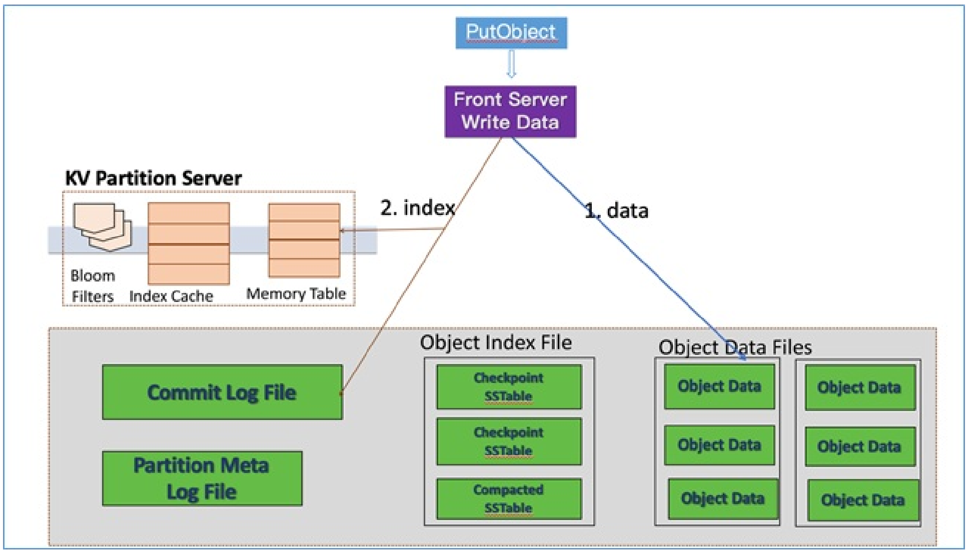


6.6.2. Data transmission process

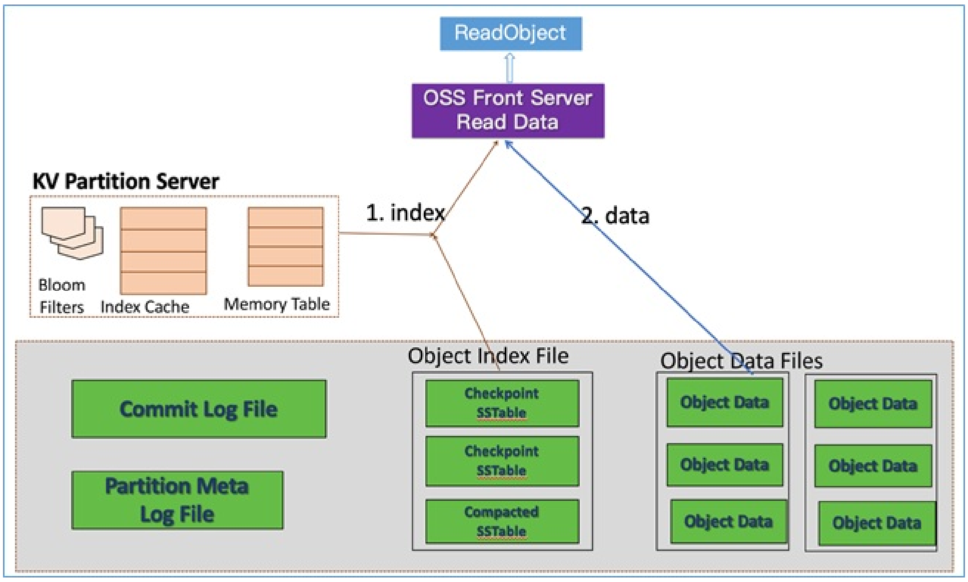
This topic describes how data is transmitted when a user accesses OSS to obtain data.

Data is transmitted in the following route during the process: `User→RESTful API→SLB-Web server (WS) →Protocol module (PM)→ Partition layer→Persistent layer` .

1. A user uses different clients such as browsers or SDKs to initiate a request that complies with the convention of OSS APIs to the OSS endpoint.
2. OSS parses the request and sends it to the LVS VIP of SLB. The backend of the LVS VIP is connected to a set of WSs. The request is forwarded to one of the WSs in the access layer.
3. The PM parses the request. First, the PM authenticates the request.
   - If the request fails the authentication, OSS returns an error code to the user.
   - If the request passes the authentication, the WS reads data from or writes data to Apsara Stack Distributed File System. The following figures show how WS writes data to and read data from Apsara Stack Distributed File System.
     - Data writing: The WS writes data to Apsara Stack Distributed File System and then updates the index of the written data.



- Data reading: The WS reads the index of the data to read from the KV server in the partition layer and caches the mapping table of the index partition. Then, the WS uses the index to read data from Apsara Stack Distributed File System and then returns the data to the user.



6.7. Best practices

6.7.1. OSS performance and scalability best practices

If you upload a large number of objects with sequential prefixes such as timestamps and letters in the object names, multiple object indexes may be stored in a single partition. If too many requests are sent to query these objects, the responsiveness may become slow. In this case, we recommend that you add random prefixes to the names of your objects.

Background information

OSS stores objects in partitions based on the UTF-8-encoded object names to process a large number of objects and high request rates. However, if you use sequential prefixes such as timestamps and letters in object names when you upload a large number of objects, multiple file indexes may be stored in a single partition. In this case, if you initiate more than 2,000 requests for the PUT, COPY, POST, DELETE, and HEAD operations per second (the number of objects on which the operations are performed indicate the number of requests), the following impacts are generated:

- The partition becomes a hotspot. The I/O capacity is exhausted, or the system automatically limits the request rate.
- OSS repartitions the data to rebalancing the data across partitions and reduce hotspots. This process may result in a longer process request time.

> Note The repartition and rebalance are performed based on the analysis result of system status and processing capability but not a fixed rule. Therefore, objects with sequential prefixes may still be stored in hotspots after repartition and rebalancing are performed.

The preceding cases affect the horizontal scalability of OSS, which degrades request rates.

To maintain the horizontal scalability and request rates, we recommend that you do not use sequential prefixes in object names. You can randomize prefix naming to evenly distribute object indexes and I/O loads to multiple partitions.

Solutions

Two methods are provided to change sequential prefixes in object names to random prefixes:

- Add a hex hash as the prefix to an object name

  If you use dates and customer IDs to generate object names, sequential prefixes with timestamps are included in object names as follows:

  ```
  sample-bucket-01/2017-11-11/customer-1/file1
  sample-bucket-01/2017-11-11/customer-2/file2
  sample-bucket-01/2017-11-11/customer-3/file3
  ...
  sample-bucket-01/2017-11-12/customer-2/file4
  sample-bucket-01/2017-11-12/customer-5/file5
  sample-bucket-01/2017-11-12/customer-7/file6
  ...
  ```

  In this case, you can calculate the MD5 hash of several characters from the customer ID as the object name prefix. If the MD5 hash of a four-character hexadecimal number is used in prefixes, the names of the objects are as follows:

  ```
  sample-bucket-01/2c99/2017-11-11/customer-1/file1
  sample-bucket-01/7a01/2017-11-11/customer-2/file2
  sample-bucket-01/1dbd/2017-11-11/customer-3/file3
  ...
  sample-bucket-01/7a01/2017-11-12/customer-2/file4
  sample-bucket-01/b1fc/2017-11-12/customer-5/file5
  sample-bucket-01/2bb7/2017-11-12/customer-7/file6
  ...
  ```

  The hash of the four-character hexadecimal number is used as the prefix. Each character can be any one of the 16 values (0-9, a-f). In the storage system, the data can be distributed to a maximum of 65,536 partitions. A maximum of 2,000 operations can be performed on each partition per second. You can determine whether the number of buckets that a hash table has meets business requirements based on the request rate.

  To list objects from the sample-bucket-01 bucket whose names contain a specified date such as 2017-11-11, you need only to list all objects from sample-bucket-01. In other words, you need only to call the ListObject operation multiple times to obtain all objects in sample-bucket-01 and list the objects whose names contain the specified date.

- Reverse the order of digits that indicate seconds in object names

  If you use the UNIX timestamps accurate to the millisecond to generate object names, sequential prefixes are included in object names as follows:

```
sample-bucket-02/1513160001245.log
sample-bucket-02/1513160001722.log
sample-bucket-02/1513160001836.log
sample-bucket-02/1513160001956.log
...
sample-bucket-02/1513160002153.log
sample-bucket-02/1513160002556.log
sample-bucket-02/1513160002859.log
...
```

In this case, you can reverse the order of the digits in the UNIX timestamp so that the object names contain no sequential prefixes. The object names after reversion are as follows:

```
sample-bucket-02/5421000613151.log
sample-bucket-02/2271000613151.log
sample-bucket-02/6381000613151.log
sample-bucket-02/6591000613151.log
...
sample-bucket-02/3512000613151.log
sample-bucket-02/6552000613151.log
sample-bucket-02/9582000613151.log
...
```

The first three digits indicate milliseconds. 1,000 values are available. The fourth digit changes every second. Likewise, the fifth digit changes every 10 seconds. Reversion greatly increases the randomness of prefixes, distributing requests evenly to each partition, reaching load balancing, and avoiding performance bottlenecks.

6.7.2. Check data transmission integrity by using CRC-64

An error may occur when data is transferred between the client and server. OSS can return the CRC-64 value of objects uploaded through any of the methods provided. The client can compare the CRC-64 value with the value calculated on the local machine to verify data integrity.

Context

OSS calculates the CRC-64 value for newly uploaded objects and stores the result as metadata of the object. OSS then adds the `x-oss-hash-crc64ecma` header to the returned response header, which indicates its CRC-64 value. This CRC-64 value is calculated based on Standard ECMA-182.

If the object exists in OSS before CRC-64 goes online, OSS does not calculate its CRC-64 value. Therefore, its CRC-64 value is not returned when the object is obtained.

Usage notes

- The PutObject, AppendObject, PostObject, and MultipartUploadPart operations return the corresponding CRC-64 value. The client can obtain the CRC-64 value returned by the server after the upload is complete and can check it against the value calculated on the local machine.
- When the MultipartComplete operation is called, the CRC-64 value of the entire object is returned if each part has a CRC-64 value. However, if a part is uploaded before the CRC-64 goes online, the CRC-64 value is returned.

- The GetObject, HeadObject, and GetObjectMeta operations return the corresponding CRC-64 value (if any). After the GetObject operation is complete, the client can obtain the CRC-64 value returned by the server and check it against the value calculated on the local machine.

  > **Note** The Get request that includes the Range header returns the CRC-64 value of the entire object.

- The newly generated object or part may not have the CRC-64 value after copy related operations such as CopyObject and UploadPartCopy are complete.

Examples

The following code in Python provides an example on how to use CRC-64 values to verify data transmission integrity.

1. Calculate the CRC-64 value.

```
import oss2
from oss2.models import PartInfo
import os
import crcmod
import random
import string
do_crc64 = crcmod.mkCrcFun(0x142F0E1EBA9EA3693L, initCrc=0L, xorOut=0xffffffffffffffffL, rev=True)
def check_crc64(local_crc64, oss_crc64, msg="check crc64"):
if local_crc64 != oss_crc64:
print "{0} check crc64 failed. local:{1}, oss:{2}.".format(msg, local_crc64, oss_crc64)
return False
else:
print "{0} check crc64 ok.".format(msg)
return True
def random_string(length):
return ''.join(random.choice(string.lowercase) for i in range(length))
bucket = oss2.Bucket(oss2.Auth(access_key_id, access_key_secret), endpoint, bucket_name)
```

2. Verify CRC-64 values for PutObject.

```
content = random_string(1024)
 key = 'normal-key'
 result = bucket.put_object(key, content)
 oss_crc64 = result.headers.get('x-oss-hash-crc64ecma', '')
 local_crc64 = str(do_crc64(content))
 check_crc64(local_crc64, oss_crc64, "put object")
```

3. Verify CRC-64 values for GetObject.

```
result = bucket.get_object(key)
 oss_crc64 = result.headers.get('x-oss-hash-crc64ecma', '')
 local_crc64 = str(do_crc64(result.resp.read()))
 check_crc64(local_crc64, oss_crc64, "get object")
```

4. Verify CRC-64 values for UploadPart and MultipartComplete.

```
part_info_list = []
key = "multipart-key"
result = bucket.init_multipart_upload(key)
upload_id = result.upload_id
part_1 = random_string(1024 * 1024)
result = bucket.upload_part(key, upload_id, 1, part_1)
oss_crc64 = result.headers.get('x-oss-hash-crc64ecma', '')
local_crc64 = str(do_crc64(part_1))
# Check whether the uploaded part_1 data is complete
check_crc64(local_crc64, oss_crc64, "upload_part object 1")
part_info_list.append(PartInfo(1, result.etag, len(part_1)))
part_2 = random_string(1024 * 1024)
result = bucket.upload_part(key, upload_id, 2, part_2)
oss_crc64 = result.headers.get('x-oss-hash-crc64ecma', '')
local_crc64 = str(do_crc64(part_2))
# Check whether the uploaded part_2 data is complete
check_crc64(local_crc64, oss_crc64, "upload_part object 2")
part_info_list.append(PartInfo(2, result.etag, len(part_2)))
result = bucket.complete_multipart_upload(key, upload_id, part_info_list)
oss_crc64 = result.headers.get('x-oss-hash-crc64ecma', '')
local_crc64 = str(do_crc64(part_2, do_crc64(part_1)))
# Check whether the final object in OSS is consistent with the local file
check_crc64(local_crc64, oss_crc64, "complete object")
```

Supported OSS SDKs

The following table describes OSS SDKs that support CRC-64 for downloads and uploads.

SDK	CRC	Sample code
OSS SDK for Java	Supported	CRCSample.java
OSS SDK for Python	Supported	object_check.py
OSS SDK for C	Supported	oss_crc_sample.c
OSS SDK for Go	Supported	crc_test.go
OSS SDK for iOS	Supported	OSSCrc64Tests.m
OSS SDK for Android	Supported	CRC64Test.java

6.8. EC storage mode

Erasure Coding (EC) is a data storage mode used by OSS. Compared with triplicate storage, EC can provide higher data reliability at lower data redundancy levels.

EC involves the following two concepts:

- Data fragments (m): Data is divided into m data fragments.
- Parity fragments (n): n parity fragments are computed based on the m data fragments.

The m data fragments and n parity fragments located on different servers compose an erasure coding group. If the number of lost data fragments is equal to or less than n, the lost segments can be restored based on the erasure coding algorithm. We recommend that you configure the value of m and n based on the number of servers.

- If you have 6 to 13 servers, we recommend that you set the values of m and n to both 2.
- If you have more than 14 servers, we recommend that you set the value of m to 8 and the value of n to 3.

Comparison between EC and triplicate storage

Compared with triplicate storage, EC is a better solution in terms of storage usage and data reliability.

Item	EC	Triplicate storage
Storage usage	`m/(m+n)` . For example, the storage usage in EC storage of the 8+3 configuration can be calculated in the following method: 8/(8+3)=72.7%	1/3=33.3%
Reliability	Allows up to n fragments to be lost. Failures on up to n servers are allowed in the worst case. For example, when m is 8 and n is 3, failures on up to three servers are allowed.	Allows up to two replicas to be lost. Failures on up to two servers are allowed in the worst case.

7.Tablestore

7.1. What is Tablestore?

7.1.1. Technical background

This topic describes the data features in the data technology (DT) era and the challenges of traditional IT software solutions for you to better understand the technical background of Tablestore.

Data features in the DT era

As the mobile Internet becomes more common and widely adopted in various industries and fields, Internet applications present the following significant features and trends:

- The amount of data that needs to be stored and processed increases exponentially. The data includes microblogs, social events, pictures, and access logs.
- As the use of mobile and Internet of Things (IoT) devices increase, the requirements for concurrent writes of structured data storage also increase.
- The data has loose schemas and tends to be semi-structured, and data fields change dynamically.
- User access features hot spots and peak hours. For example, during promotional activities, user access soars within a few minutes.
- The mobile Internet allows users to connect to Internet applications at any time. Service instability caused by failures or even planned service failures greatly affects user experience. Therefore, high availability is required.
- Large amounts of data increase the requirements for the performance and scale of computing and analysis.

Challenges to traditional IT software solutions

Traditional IT software solutions face the following trends and challenges:

- Scalability

  Traditional software such as relational databases are incapable of handling such fast-growing data. It bottlenecks data write throughput and access efficiency. In traditional database solutions, databases and tables are manually and statically partitioned. This method requires large amounts of maintenance. In particular scenarios where nodes are added to increase the storage capacity, you must repartition and migrate existing data. During this process, it is difficult to guarantee service performance, stability, and availability. The whole process is complex.

- Data model changes

  Data in traditional databases is processed based on a schema. The number of columns for data storage is fixed and seldom modified. Frequent changes to the table schema and column count affect service availability. Therefore, traditional solutions are incapable of handling the increasing volumes of loosely structured data from Internet applications.

- Quick scaling

In traditional solutions, business access loads are stable, and the system is not required to scale resources in a short time. When resources need to be scaled, a large amount of labor is required to reparation and migrate data. Then, when business loads decline, the hosts added during scaling must be removed to avoid low resource usage, and data must be migrated again. This process is complex and inefficient.

- O&M guarantees

  In traditional software solutions, services are recovered when hardware (network devices or disks) failures occur. You must manually replace hardware, upgrade software, and configure tuning and updates. To ensure that applications are not aware of these processes and avoid deterioration of service availability, a special engineering team is required to implement O&M. Therefore, workloads caused from recruitment and fund investment bring a huge challenge to fast-developing enterprises.

- Computing bottlenecks

  The current business system uses Online Transaction Processing (OLTP) to process and analyze data in relational databases such as MySQL and Microsoft SQL Server. These relational databases are used to process transactions. Consistency and atomicity are maintained while data is frequently inserted and modified. However, if the amount of data that needs to be queried or calculated is too large, such as tens of millions or even billions of records of data, or if the computing is complex, the OLTP databases cannot meet the requirements.

7.1.2. Tablestore technologies

To improve scalability, Tablestore partitions tables and schedules data partitions to different nodes. When hardware failures occur on a single server, Tablestore uses heartbeat mechanism to find the node where failures occur. The partition in the node is migrated to a normal node, and the service is recovered and continues.

Data partitioning and load balancing

The first column of a primary key in each row of a table is the partition key. The system splits a table into multiple partitions based on the value of the partition key. These partitions are evenly scheduled across different storage nodes. When the data in a partition exceeds the limit size, the partition is split into two smaller partitions. The data and access loads are distributed to these two partitions. The partitions are scheduled to different nodes. As a result, access loads are scattered to different nodes. Eventually, the single-table data scale and access loads can be linearly scaled.

Technical indicator: Tablestore can store petabytes of data in a single table and allows you to simultaneously read/write millions of data.

Automatic recovery from single points of failure (SPOFs)

Each node in the storage engine of Tablestore provides services for multiple data partitions of different tables. The master node manages partition distribution and scheduling, and also monitors the health of each service node. If a service node fails, the master node migrates data partitions from the faulty node to other healthy nodes. The migration is logically performed, and does not involve physical entities. Therefore, services can rapidly recover from SPOFs.

Technical indicator: SPOFs affect services of only a part of data partitions and services can recover within single-digit minutes.

Zone-disaster recovery and geo-disaster recovery

To meet business security and availability requirements, Tablestore provides zone-disaster recovery and geo-disaster recovery based on primary and secondary clusters. Disaster recovery supports instance-based recovery. Any table operation on the primary instance, including insertion, update, or deletion, is synchronized to the table of the same name in the secondary instance. The duration of data synchronization between the primary and secondary instances depends on the network environment of the primary and secondary clusters. In the ideal network environment, the synchronization latency is within single-digit milliseconds. Before the manual failover, you must stop resource access to the primary cluster and wait for all data to be completely backed up. You can perform only one failover in an hour. After the failover, data in the original cluster is deleted, and a secondary cluster is configured.

In zone-disaster recovery based on primary and secondary clusters, the endpoints remain unchanged when applications access Tablestore in the primary and secondary clusters. In other words, the application endpoints do not need to be changed after the failover. In geo-disaster recovery based on primary and secondary clusters, the endpoints of the primary and secondary clusters are different. After the failover, endpoints need to be changed for applications.

Technical indicator: The RTO of Tablestore is smaller than 2 minutes, the RPO is smaller than 5 minutes, and the RCO is 1.

7.2. Benefits

Tablestore provides the following benefits:

Scalability

- Tablestore imposes no upper limit on the amount of data that can be stored in tables. When the amount of data increases, Tablestore adjusts partitions to provide more storage space for tables and improve the capability of handling access request bursts.
- Tablestore supports CPUs, disks, memory, and network interface controllers (NICs) of different specifications in a single-component cluster without affecting cluster running performance. This ensures maximum compatibility with existing devices.

High performance

If you use a high-performance instance, its average access latency of single rows is measured in single-digit milliseconds. The read/write performance is not affected by the size of data in a table.

Data reliability

- Tablestore provides high data reliability. It stores multiple data copies and restores data when any of the copies become invalid.
- Tablestore supports automatic fault tolerance for the failures of server hard disks in a cluster and supports hot swapping of hard disks. In the event of a hard disk failure, services can be restored within two minutes.
- Tablestore supports full or incremental backup and data recovery from storage.
- Tablestore supports the backup between data clusters in different data centers. The backup process is visualized.
- Tablestore supports the backup and restoration of the metadata, files, and tables of key components.

High availability

Tablestore uses automatic failure detection and data migration to shield applications from host- and network-related hardware faults, which provides high availability for your applications.

Ease of management

- Tablestore automatically performs complex O&M tasks, such as the management of data partitions, software and hardware upgrades, configuration updates, and cluster scale-out.
- You can store audit logs to Log Service and download logs from Log Service. This facilitates the long-term storage and management of audit logs.

Access security

- Tablestore provides multiple permission management mechanisms. It verifies and authenticates the identity of each application request to prevent unauthorized data access, which improves data security.
- Tablestore supports the management of data access permissions, including logon permissions, table creation permissions, read and write permissions, and whitelist-related permissions.
- Tablestore allows you to use the Apsara Uni-manager Management Console to manage administrative permissions, including administrator classification. You can use the Apsara Uni-manager Management Console to manage user permissions in a centralized manner. You can manage the access control features of all components in the system. You can also block common users from querying access control details and simplify access control for administrators. This improves the usability of access control.

Strong consistency

Tablestore ensures high data consistency for data writes. After a write operation succeeds, three replicas are written to a disk. Applications can read the latest data immediately.

Flexible data models

Tablestore tables do not require a fixed format. Each row can contain a different number of columns. Tablestore supports multiple data types, including Integer, Boolean, Double, String, and Binary.

Monitoring integration

You can log on to the Tablestore console to obtain monitoring information in real time, including the requests per second and average response latency.

Multitenancy

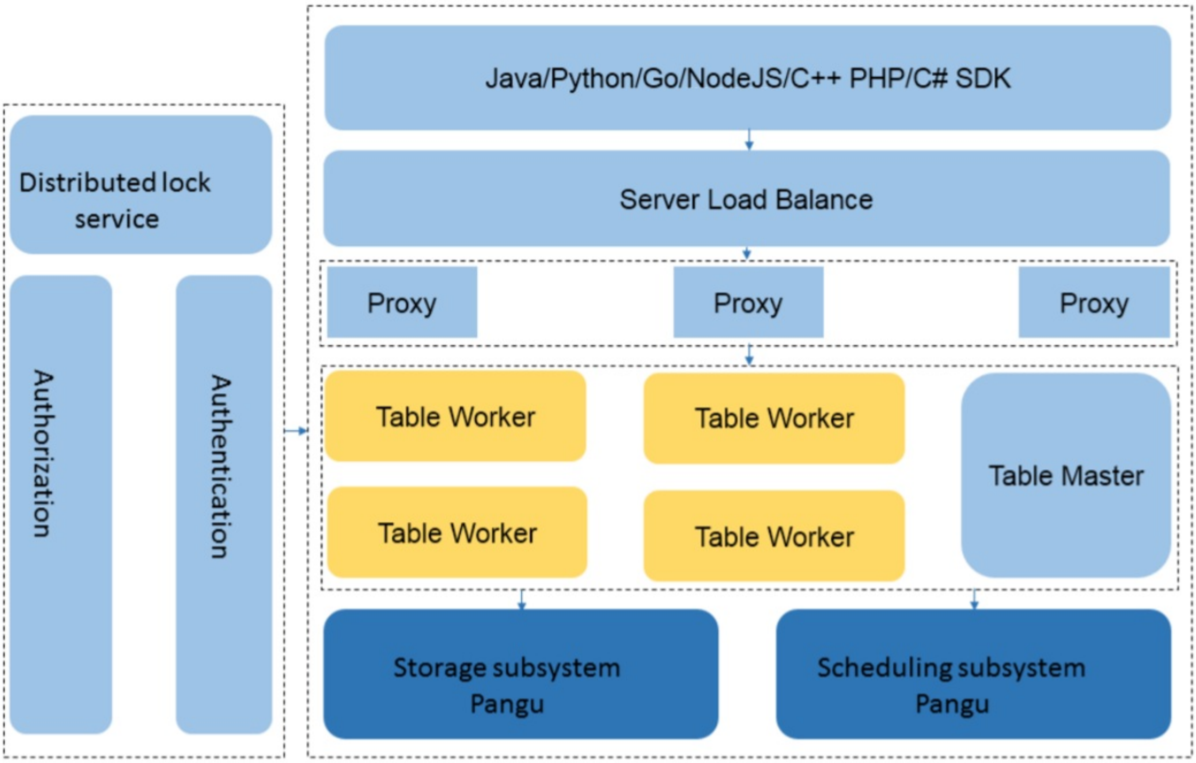
- Isolation: allows tasks of multiple tenants (projects) to be submitted to different queues and run separately. Resources are isolated among tenants.
- Permission: allows you to manage tenants in a centralized manner, dynamically configure and manage tenant resources, isolate resources, view statistics for resource usage, and manage tenants at multiple levels in the console.
- Scheduling: supports multi-tenant scheduling of multiple clusters and multiple resource pools.

7.3. Architecture

This topic describes the Tablestore architecture.

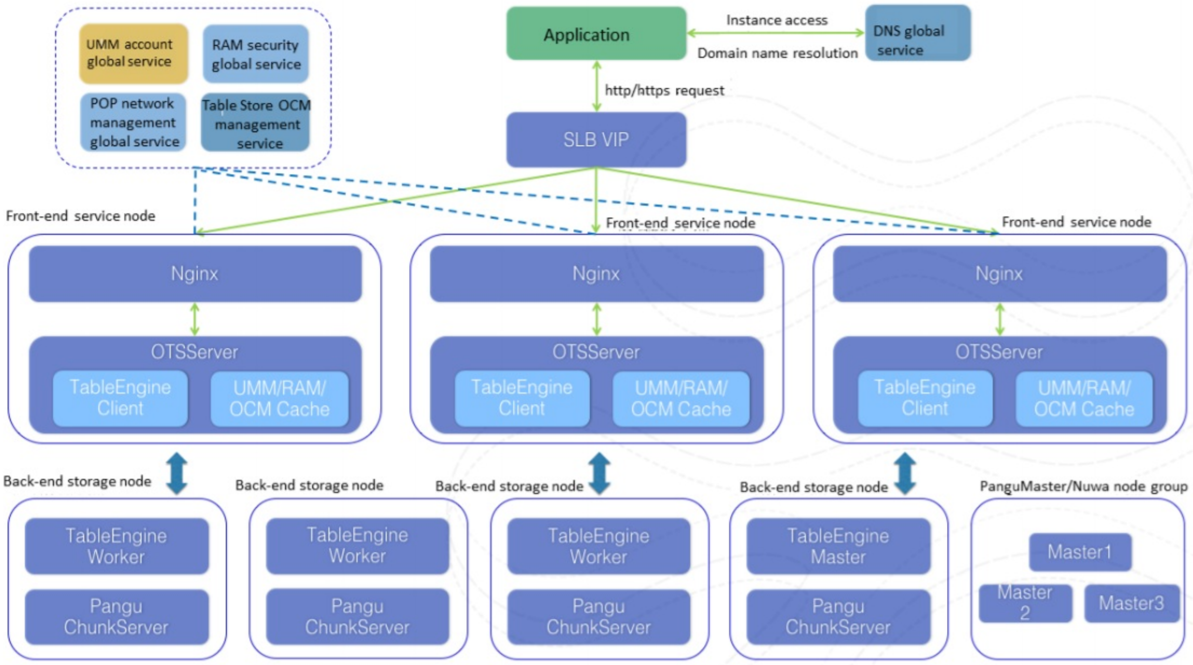
The architecture of Tablestore is referenced from Bigtable (one of the three core technologies of Google) and uses the log-structured merge-tree (LSM) storage engine to provide high write performance. The performance of primary key-based single-row queries and range queries is stable and predictable. The performance is not affected by the volume of data and access concurrency.

The following figure shows the basic architecture of Tablestore.



- The top layer is the protocol access layer. Server Load Balancer (SLB) distributes user requests to various proxy nodes. The proxy nodes receive requests that are sent by using the RESTful protocol and implement security authentication.
  - If the authentication succeeds, the user requests are forwarded to the corresponding data engine based on the value of the first primary key column for further operations.
  - If the authentication fails, error information is returned to the user.
- Table Worker is the data engine layer that processes structured data. It uses a primary key to search for or store data. Table Worker supports large-scale access request bursts.
- The bottom layer is the persistent storage layer. Apsara Distributed File System is deployed at this layer. Metadata is stored on masters. A distributed message consistency protocol (or Paxos) is adopted between masters to ensure the metadata consistency. This way, efficient distributed file storage and access are achieved. This method ensures that three copies of data are stored in the system and that the system can recover from any hardware or software fault.

The following figure shows the detailed architecture of Tablestore.

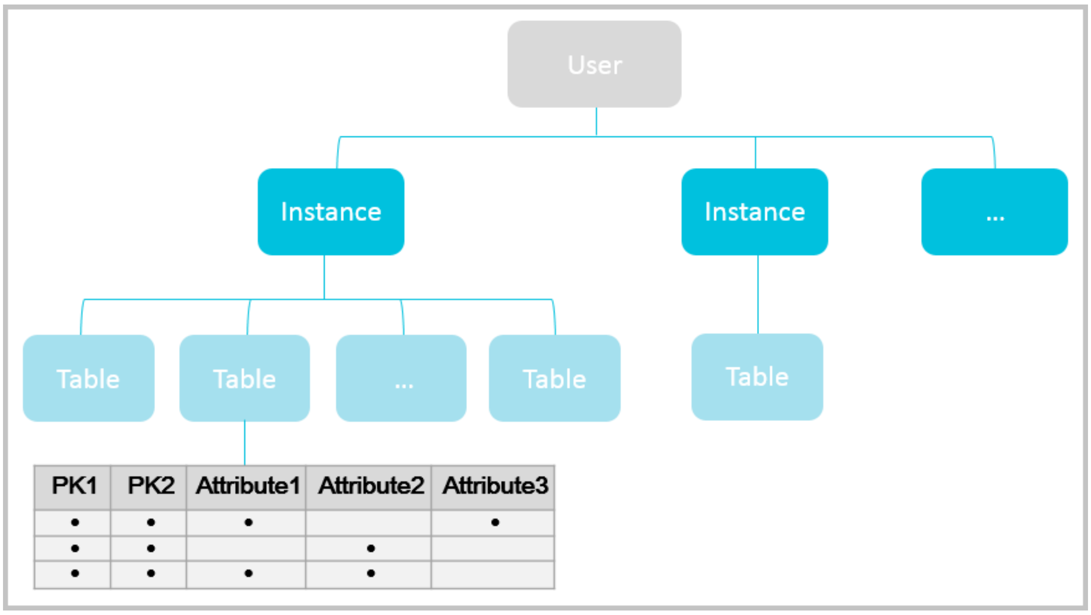


7.4. Features

7.4.1. Users and instances

This topic describes the architecture of users and instances.

The following figure shows the Tablestore architecture in relation to a user and instances.



PK1	PK2	Attribute1	Attribute2	Attribute3
•	•	•		•
•	•		•	
•	•	•	•	

- Users can log on with an Apsara Stack tenant account.
- User operations can be audited in fine granularity.
- Users organize resources based on instances. A user can create multiple instances and use each

instance to create and manage multiple data tables.

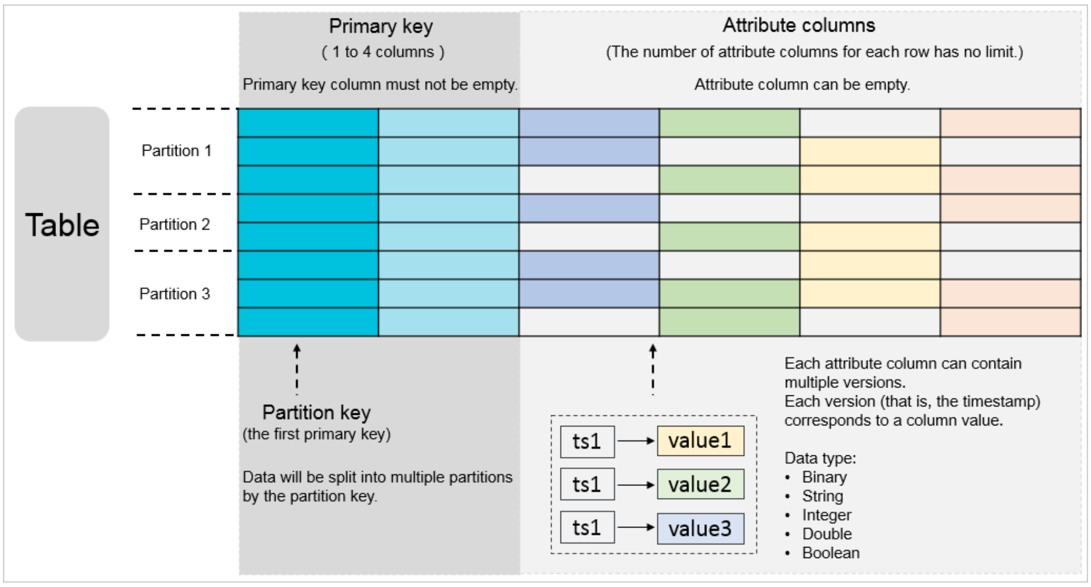
- An instance is the basic unit of multi-tenant isolation.
- User permissions can vary based on their roles.

7.4.2. Data tables

This topic describes the structure of data tables.

The following figure shows the data table structure.

Data table structure


Primary key
(1 to 4 columns)
Primary key column must not be empty.
Attribute columns
(The number of attribute columns for each row has no limit.)
Attribute column can be empty.
Table
Partition 1
Partition 2
Partition 3
Partition key
(the first primary key)
Data will be split into multiple partitions by the partition key.
ts1
value1
ts1
value2
ts1
value3
Each attribute column can contain multiple versions.
Each version (that is, the timestamp) corresponds to a column value.
Data type:
• Binary
• String
• Integer
• Double
• Boolean

- A data table is the basic unit of resource allocation.
- A table is a collection of rows. A row consists of primary key columns and attribute columns.
- A table partitions data based on the value of the first primary key column.
- All rows in a table must have the same quantity of primary key columns that share the same names.
- The quantity, names and data types of attribute columns in a row can be different.
- The number of attribute columns contained in a row is not limited. However, the maximum number of attribute columns that can be written in each request is 1,024.
- A table can contain at least hundreds of billions of rows of data.
- A table can store petabytes of data.

7.4.3. Data partitioning

This topic describes the features of data partitioning.

- A table partitions data based on the value of the first primary key column.
- The rows whose first primary key column values are within the same partition key value range are allocated to the same partition.
- To improve load balancing, Tablestore splits and merges partitions based on specific rules.
- We recommend that you do not store more than 10 GB of data in rows that share the same partition key.

7.4.4. Common commands and functions

This topic describes common commands used to manage tables and common functions used to manage data in tables.

Common commands used to manage tables

- ListTable: lists all tables in an instance.
- CreateTable: creates a table.
- DeleteTable: deletes a table.
- DescribeTable: queries the attributes of a table.
- UpdateTable: updates the reserved read/write throughput configuration of a table.
- ComputeSplitPointsBySize: logically partitions all table data into multiple partitions of a specified size, and returns the split points between these partitions and the prompt of the hosts where partitions reside.

Common functions used to manage data in tables

- GetRow: reads data from a single row.
- PutRow: inserts a row of data.
- UpdateRow: updates a row of data.
- DeleteRow: deletes a row of data.
- BatchGetRow: reads multiple rows in one or more tables simultaneously.
- BatchWriteRow: inserts, updates, or deletes multiple rows in one or more tables.
- GetRange: reads reads a range of data from a table.

7.4.5. Authorization and access control

This topic describes the access control for Tablestore and the operations you can perform in the Apsara Uni-manager Management Console.

Tablestore permissions

Tablestore integrates RAM and VPC to support the following access control mechanisms:

- Table-level authorization
- Operation-level access control
- Authentication based on IP address limits, HTTPS, multi-factor authentication (MFA), and access time limits
- Temporary access authorization based on STS
- VPC-based access control

Operations in the Apsara Uni-manager Management Console

- Account logons and authentication
- Instance creation, management, and deletion in GUI
- Table creation, management, deletion, and reserved read/write throughput adjustment in GUI
- Monitoring information displaying based on tables

8.ApsaraDB RDS

8.1. 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 database engines, which are MySQL, SQL Server, PolarDB, and PostgreSQL. You can create database instances based on these engines to meet your business requirements.

ApsaraDB RDS for 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 for MySQL 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.

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, TDE, data migration, and management for instances, accounts, and databases.

PolarDB

PolarDB is a stable, secure, and scalable enterprise-class relational database service. Based on PostgreSQL, PolarDB enhances performance, application solutions, and compatibility. PolarDB also provides the capability of directly running Oracle applications. You can run a variety of enterprise applications on PolarDB stably at low costs.

PolarDB provides features such as account management, resource monitoring, backup and restoration, and security control. These features are under continuous improvement, and more features are under development to adapt to PolarDB.

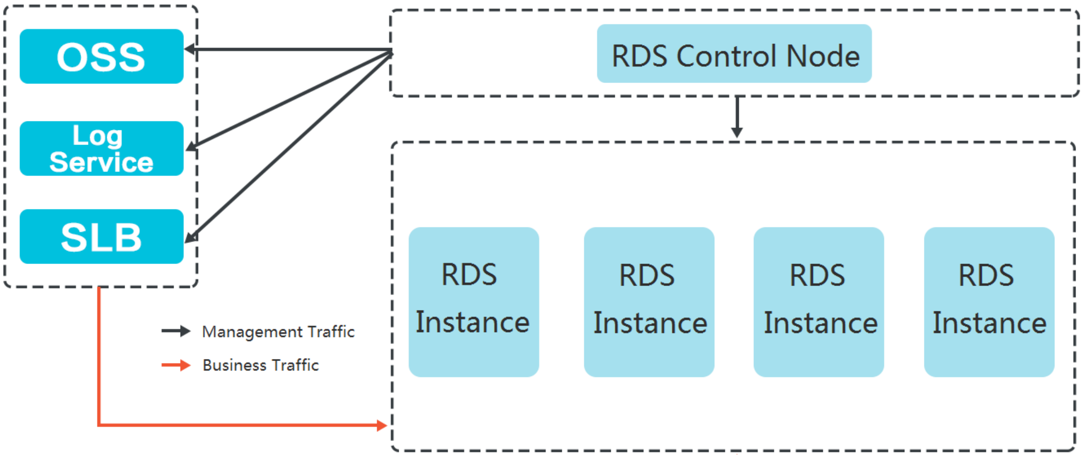
ApsaraDB RDS for PostgreSQL

ApsaraDB RDS for PostgreSQL is an advanced open source database service that is fully compatible with SQL and supports a diverse range of data formats such as JSON, IP, and geometric data. In addition to support for 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 operations and maintenance loads.

8.2. Architecture

The following figure shows the system architecture of ApsaraDB RDS.

ApsaraDB RDS system architecture

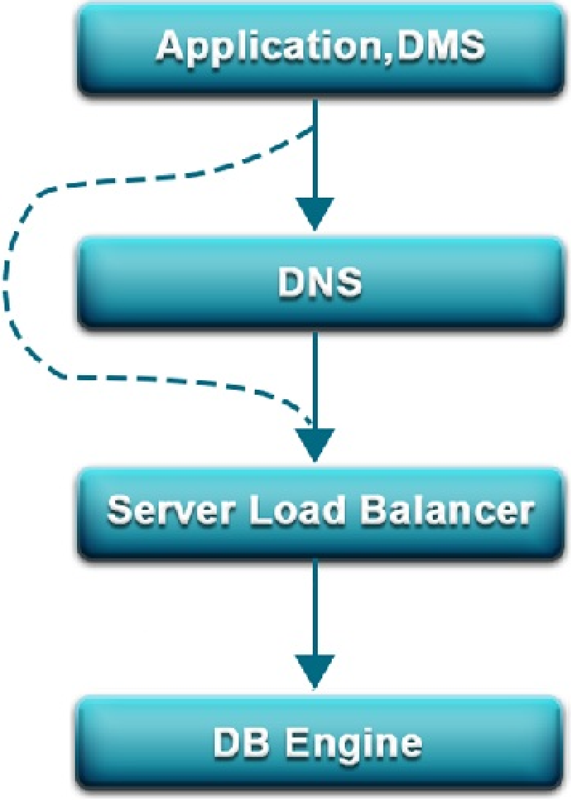


8.3. Features

8.3.1. Data link service

The data link service allows you to add, delete, modify, and query the table schema and data.

ApsaraDB RDS data link service



DNS

The DNS module can dynamically resolve domain names to IP addresses. Therefore, IP address changes do not affect the performance of ApsaraDB RDS instances.

For example, assume that the domain name of an ApsaraDB RDS instance is test.rds.aliyun.com, and its corresponding IP address is 10.1.1.1. The instance can be accessed when test.rds.aliyun.com or 10.1.1.1 is configured in the connection pool of a program.

After this ApsaraDB RDS instance is migrated or its version is upgraded, the IP address may change to 10.1.1.2. If the domain name test.rds.aliyun.com is configured in the connection pool, the instance can still be accessed. However, if the IP address 10.1.1.1 is configured in the connection pool, the instance is no longer accessible.

SLB

The Server Load Balancer (SLB) module provides both the internal and public IP addresses of an ApsaraDB RDS instance. Therefore, server changes do not affect the performance of the instance.

For example, assume that the internal IP address of an ApsaraDB RDS instance is 10.1.1.1, and the corresponding Proxy or DB Engine runs on 192.168.0.1. The SLB module typically redirects all traffic destined for 10.1.1.1 to 192.168.0.1. If 192.168.0.1 fails, another server in the hot standby state with the IP address 192.168.0.2 takes over for the initial server. In this case, the SLB module redirects all traffic destined for 10.1.1.1 to 192.168.0.2, and the ApsaraDB RDS instance continues to provide services normally.

DB Engine

The following table describes the major database protocols supported by ApsaraRD RDS.

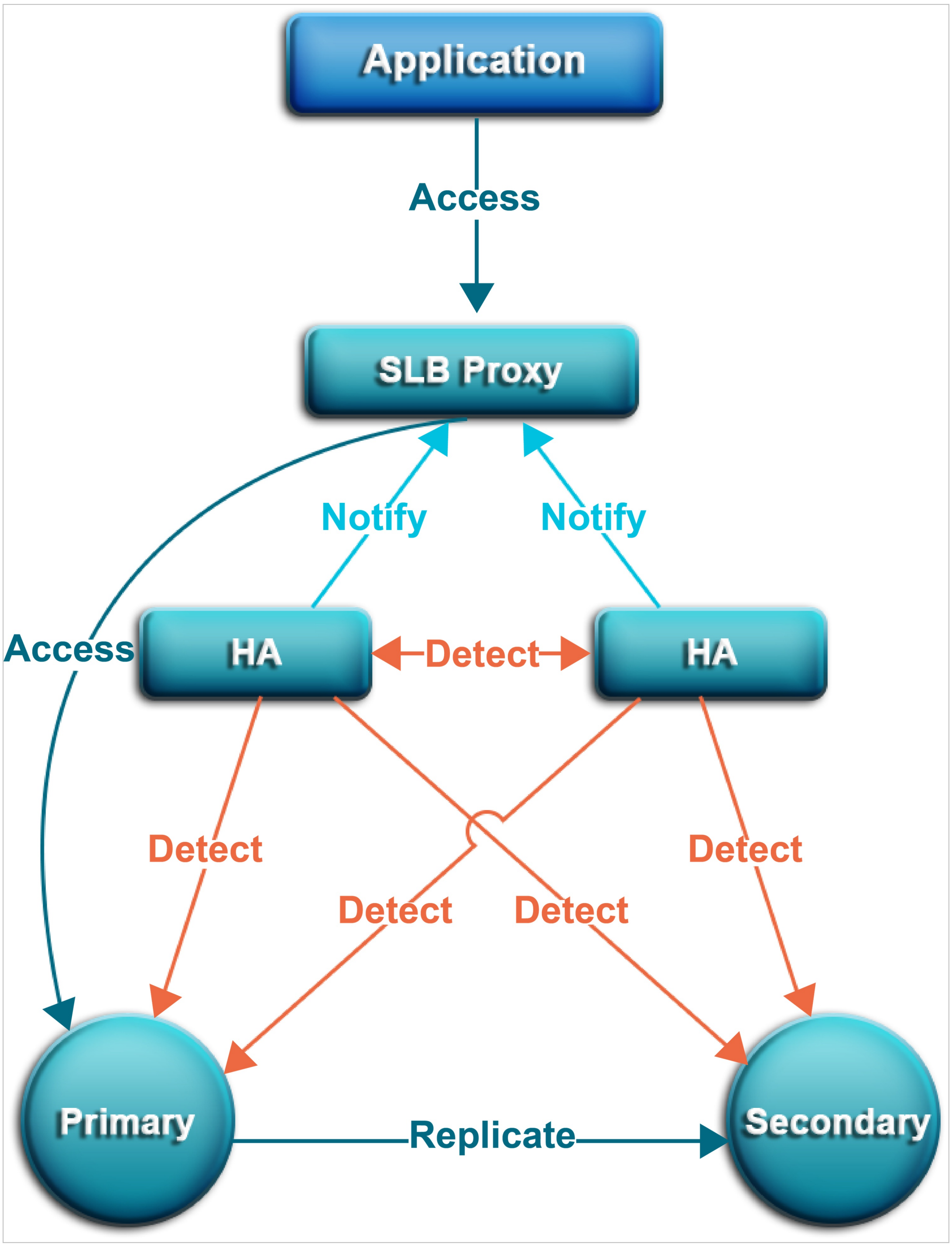
ApsaraRD RDS database protocols

RDBMS	Version
MySQL	5.6 and 5.7
SQL Server	2012, 2016, and 2017
PostgreSQL	9.4, 10, 11, and 12
PolarDB	11

8.3.2. High-availability service

The high-availability (HA) service ensures the availability of data link services and processes internal database exceptions. The HA service is implemented by multiple HA nodes.

ApsaraDB RDS HA service



Detection

The Detection module checks whether the primary and secondary nodes of the DB Engine are providing services normally.

The HA node uses heartbeat information taken at 8 to 10 second intervals to determine the health status of the primary node. This information, along with the health status of the secondary node and heartbeat information from other HA nodes, provides a reference for the Detection module. All this information helps the module avoid misjudgment caused by exceptions such as network jitter. Failover can be completed within a short time.

Repair

The Repair module maintains the replication relationship between the primary and secondary nodes of the DB Engine. It can also correct errors that occur on the nodes during normal operations. For example:

- It can automatically restore primary/secondary replication after a disconnection.
- It can automatically repair table-level damage to the primary or secondary node.
- It can save and automatically repair the primary or secondary node when the node fails.

Notice

The Notice module informs the Server Load Balancer (SLB) or Proxy module of status changes to the primary and secondary nodes to ensure that you always access the correct node.

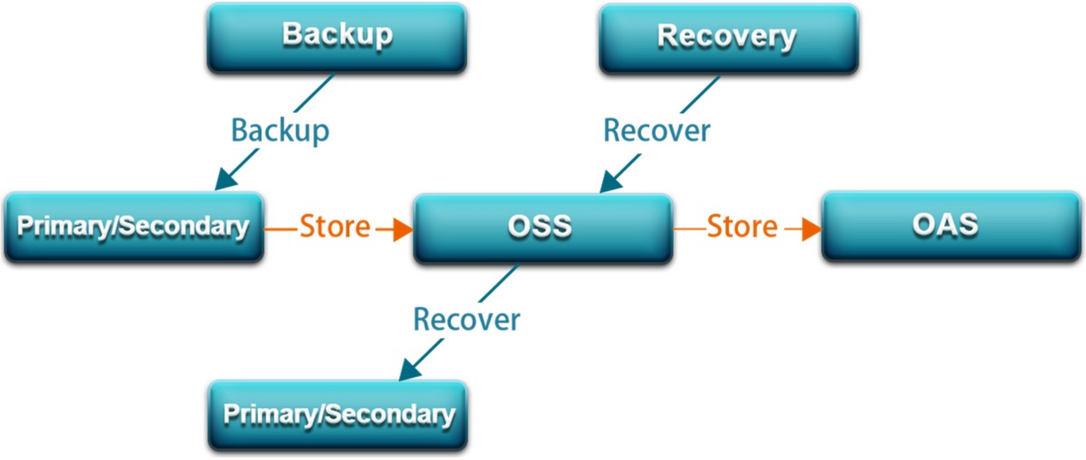
For example, the Detection module discovers problems with the primary node and instructs the Repair module to resolve these problems. If the Repair module fails to resolve a problem, it instructs the Notice module to perform traffic switchover. The Notice module forwards the switching request to the SLB or Proxy module. Then, all traffic is redirected to the secondary node.

Meanwhile, the Repair module creates a new secondary node on a different physical server and synchronizes this change back to the Detection module. The Detection module rechecks the health status of the instance.

8.3.3. Backup service

The backup service supports offline data backup, storage, and recovery.

ApsaraDB RDS backup service



Backup

The Backup module compresses and uploads data and logs on both the primary and secondary nodes. ApsaraDB RDS uploads backup files to Object Storage Service (OSS) and dumps the backup files to a more cost-effective and persistent Archive Storage system. When the secondary node operates normally, backups are always created on the secondary node. This way, the services on the primary node are not affected. When the secondary node is unavailable or damaged, the Backup module creates backups on the primary node.

Recovery

The Recovery module restores backup files from OSS to a destination node. The Recovery module provides the following features:

- Primary node rollback: rolls back the primary node to a specific point in time when an operation error occurs.
- Secondary node repair: creates a new secondary node to reduce risks when an irreparable fault occurs on the secondary node.
- Read-only instance creation: creates a read-only instance from backup files.

Storage

The Storage module uploads, dumps, and downloads backup files.

All backup data is uploaded to OSS for storage. You can obtain temporary links to download the data.

In specific scenarios, the Storage module allows you to dump backup files from OSS to Archive Storage for more cost-effective and longer-term offline storage.

8.3.4. Monitoring service

ApsaraDB RDS provides multilevel monitoring services across the physical, network, and application layers to ensure service availability.

Service

The Service module tracks the status of services. For example, the Service module monitors whether Server Load Balancer (SLB), Object Storage Service (OSS), and other cloud services on which ApsaraDB RDS depends are operating normally. The monitored metrics include functionality and response time. The Service module also uses logs to determine whether the internal services of ApsaraDB RDS are operating properly.

Network

The Network module tracks statuses at the network layer. The following metrics are monitored:

- Connectivity between Elastic Compute Service (ECS) and ApsaraDB RDS
- Connectivity between physical servers of ApsaraDB RDS
- Rates of packet loss on vRouters and vSwitches

OS

The OS module tracks the statuses of hardware and OS kernel. The following metrics are monitored:

- Hardware maintenance: The OS module constantly checks the operating status of the CPU, memory, motherboard, and storage device. It can predict faults in advance and automatically submit repair reports when it determines a fault is likely to occur.
- OS kernel monitoring: The OS module tracks all database calls and analyzes the causes of slow calls or call errors based on the kernel status.

Instance

The Instance module collects the following information about ApsaraDB RDS instances:

- Instance availability information
- Instance capacity and performance metrics

- Instance SQL execution records

8.3.5. Scheduling service

The scheduling service allocates resources and manages instance versions.

Resource

The Resource module allocates and integrates underlying ApsaraDB RDS resources when you activate and migrate instances. When you create an instance by using the ApsaraDB RDS console or an API operation, the Resource module calculates the most suitable host to carry traffic to and from the instance. A similar process occurs when ApsaraDB RDS instances are migrated.

After instances are repeatedly created, deleted, or migrated, the Resource module calculates the degree of resource fragmentation. In addition, it integrates resources on a regular basis to improve the service carrying capacity.

8.3.6. Migration service

The migration service can migrate data from your self-managed databases to ApsaraDB RDS.

DTS

DTS can migrate data from your self-managed databases to ApsaraDB RDS for MySQL without the need to stop services.

DTS is a data exchange service that streamlines data migration, real-time synchronization, and subscription. DTS is dedicated to implementing remote and millisecond-speed asynchronous data transmission in various scenarios. Based on the active geo-redundancy architecture designed for Double 11, DTS can make the data architecture secure, scalable, and highly available by providing real-time data streams to up to thousands of downstream applications.

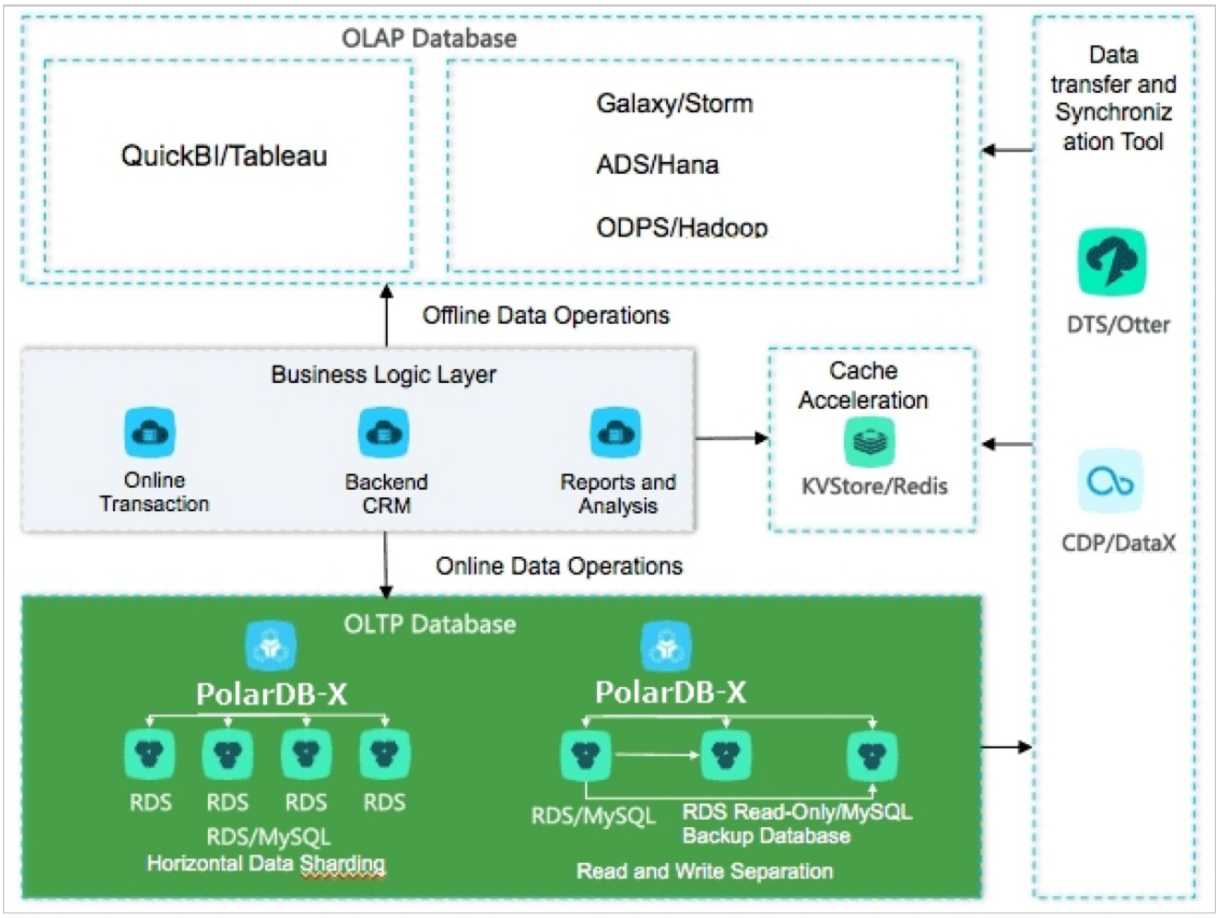
9.Cloud Native Distributed Database PolarDB-X

9.1. What is PolarDB-X?

Cloud Native Distributed Database PolarDB-X is a middleware service independently developed by Alibaba Group for scale-out of single-instance relational databases. It is compatible with Distributed Relational Database Service (DRDS).

PolarDB-X is the standard of relational database access for Alibaba Group. It shares the database sharding logic with Taobao Distributed Data Layer (TDDL). Compatible with the MySQL protocol, PolarDB-X supports most MySQL data manipulation language (DML) and data definition language (DDL) syntax. It provides the core capabilities of distributed databases, such as database sharding, table sharding, smooth scale-out, configuration changing, and transparent read/write splitting. It is lightweight (stateless), flexible, stable, and efficient, and provides you with O&M capabilities throughout the lifecycle of distributed databases.

PolarDB-X is mainly used for operations on large-scale online data, which focuses on frontend businesses for writing data to databases. By splitting data in specific business scenarios, it maximizes operation efficiency to meet the requirements of high-concurrency and low-latency database operations.



PolarDB-X mainly solves the following problems:

- Capacity bottleneck of single-instance databases: As the data volume and access volume increase, traditional single-instance databases encounter great challenges that cannot be completely solved by hardware upgrades. In distributed database solutions of PolarDB-X, multiple instances work jointly, which effectively resolves the bottlenecks of data storage capacity and access volumes.
- Difficult scale-out of relational databases: Due to the inherent attributes of distributed databases, data can be stored to different shards in PolarDB-X through smooth data migration, supporting the dynamic scale-out of relational databases.

9.2. Benefits

Distributed architecture

The distributed architecture of Cloud Native Distributed Database PolarDB-X allows for horizontal partitioning of data and the cluster deployment of a single service. This resolves the single-instance bottlenecks of Server Load Balancer (SLB), PolarDB-X, and ApsaraDB RDS for MySQL and facilitates service scalability.

Elastic scaling

PolarDB-X instances and ApsaraDB RDS for MySQL instances can be dynamically added and removed for flexible service capabilities.

High performance

PolarDB-X can be integrated with ApsaraDB RDS for MySQL to split data in specific business scenarios and cluster data based on major operations. This helps speed up the response to online transaction operations.

Security and controllability

PolarDB-X supports an account and permission system similar to that of single-instance databases. PolarDB-X provides useful features, such as the IP address whitelist and the default disabling of high-risk SQL statements. PolarDB-X provides a complete standard API system that can be integrated with your on-premises management system. A complete portfolio of product support and architecture services is also available to you.

9.3. Architecture

PolarDB-X supports two output methods: integrated output by Apsara Stack and separate output by Alibaba middleware. The two output methods differ in the features and the components that PolarDB-X depends on.

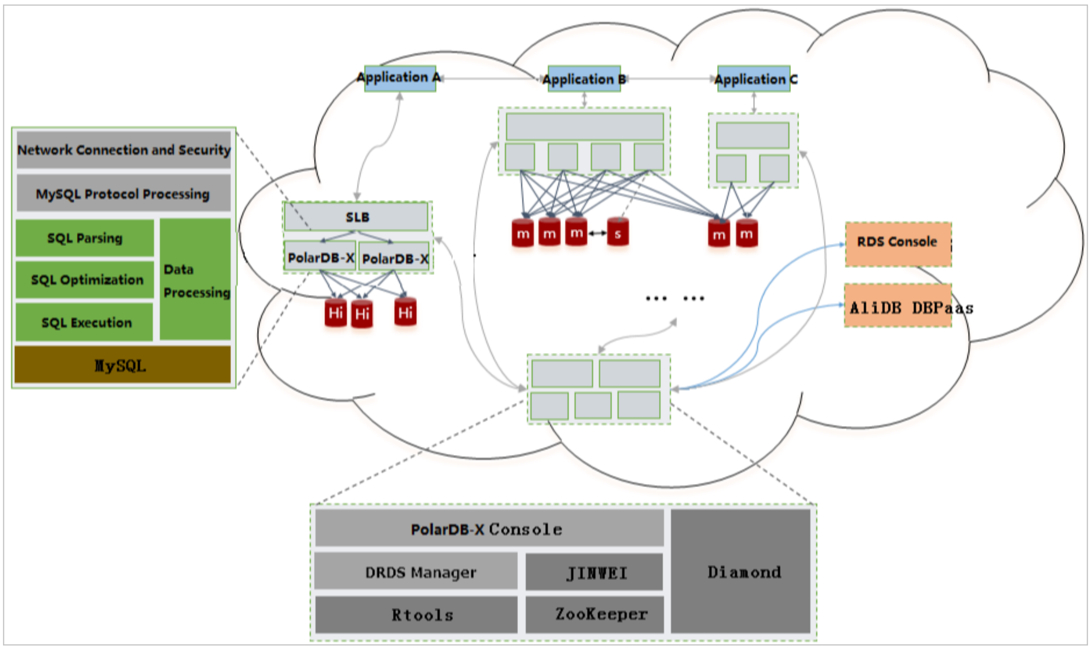
The following table describes the differences between the two methods.

Difference	Integrated output by Apsara Stack	Separate output by Alibaba middleware
MySQL	RDS for MySQL	Alibaba Group Database Platform as a Service (DBPaaS)
Load balancing	Centralized Server Load Balancer (SLB)	Client-side load balancer (VIPServer)

Difference	Integrated output by Apsara Stack	Separate output by Alibaba middleware
Support for special storage	None	High compression-ratio column store (HiStore)

The following figure shows the system architecture of PolarDB-X.

PolarDB-X system architecture



PolarDB-X Server

PolarDB-X Server is the service layer of PolarDB-X. Multiple service nodes constitute a service cluster to provide distributed database services, including read/write splitting, routed SQL execution, result combination, dynamic database configuration, and globally unique identifiers (GUIDs).

Note PolarDB-X instances are stateless nodes. PolarDB-X uses ApsaraDB RDS for MySQL instances for data storage. PolarDB-X encrypts data by using encryption algorithms, including transparent data encryption (TDE) supported by ApsaraDB RDS for MySQL.

High-availability clusters

A redundancy design is adopted for each system component to prevent single point of failures (SPOFs) after a node fails.

ApsaraDB RDS for MySQL (represented by m and s in the figure)

ApsaraDB RDS for MySQL stores data and performs online data operations. It achieves high availability by using MySQL primary/secondary replication. ApsaraDB RDS for MySQL also enables dynamic database failover by using the primary/secondary switchover mechanism.

In the ApsaraDB RDS for MySQL console, you can manage and monitor instances, and manage alerts and resources throughout the instance lifecycle.

HiStore

When PolarDB-X outputs data separately (not overall output by Apsara Stack), it uses HiStore as the physical storage. HiStore is a low-cost, high-performance database developed by Alibaba to support column store. By using the column store, knowledge grid, and multiple cores, HiStore provides higher data aggregation and ad hoc query capabilities, with lower costs than row store (such as MySQL).

You can implement management, monitoring, and alerting within the instance lifecycle in the HiStore console.

DBPaaS

If you select the separate output of PolarDB-X instead of the integrated output by Apsara Stack, you can perform operations on the built-in MySQL O&M platform DBPaaS. For example, you can manage and monitor instances, and manage alerts and resources throughout the instance lifecycle.

Centralized SLB

You do not need to install a client on your instance. SLB is used to distribute your requests. When an instance fails or a new instance is added, SLB ensures that traffic on the storage nodes is evenly distributed.

VIPServer that enables client-side load balancing

You must install a client on your instance to reduce dependency on the central controller. This indicates that interaction is performed only when the load configuration changes. VIPServer is used to distribute the traffic of your requests. When an instance fails or a new instance is added, VIPServer ensures that traffic on the storage nodes is evenly distributed.

Diamond

Diamond is a system that allows you to configure storage and manage instances for PolarDB-X. You can configure storage settings, query instance information, and subscribe to notifications. In PolarDB-X, Diamond stores the source data of databases and configurations that include sharding rules and PolarDB-X switches.

Data Replication System

Data Replication System migrates and synchronizes data for PolarDB-X. Its core capabilities include full data migration and incremental data synchronization. Its derived capabilities include smooth data import, smooth scale-out, and global secondary indexes. Data Replication System requires the support of ZooKeeper and PolarDB-X Rtools.

PolarDB-X Console for user operations

PolarDB-X Console is designed for database administrators (DBAs) to isolate resources and operations based on users. It enables multiple features, such as instance management, database and table management, read/write splitting configuration, smooth scale-out, monitoring display, and the IP address whitelist.

PolarDB-X Manager for O&M operations

PolarDB-X Manager is designed for global O&M operations and DBAs. It enables resource management and system monitoring for PolarDB-X. The core features contain the following two aspects:

- Manage all the resources on which ApsaraDB RDS for MySQL instances depend, including virtual machines, load balancers, and domain names.

- Monitor the PolarDB-X instance status, including the queries per second (QPS), active threads, the number of connections, node network I/O, and node CPU utilization.

Rtools

Rtools is the O&M support system of PolarDB-X. It allows you to manage database configurations, read/write weights, connection parameters, topologies of databases and tables, and sharding rules.

9.4. Features

9.4.1. Horizontal partitioning (sharding)

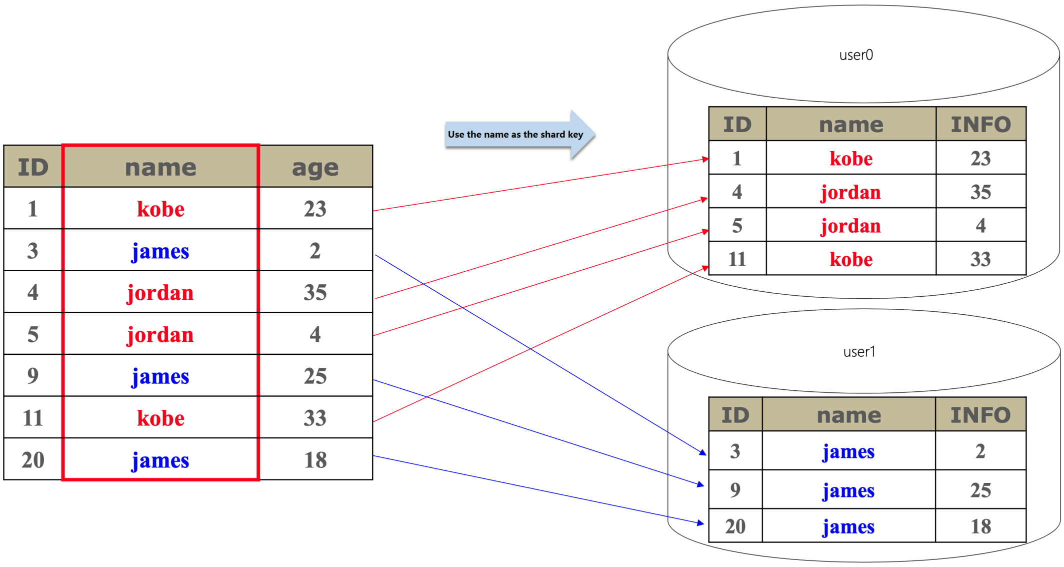
The core principle of PolarDB-X is horizontal partitioning of data, where data in a logical database is distributed and stored to multiple stable MySQL databases according to certain rules. These MySQL databases can be distributed across multiple instances or even across data centers, but provide external services (add, delete, modify, and query operations) as a single MySQL database. After partitioning, a physical database on an MySQL instance is called a database shard and a physical table is called a table shard (each table shard is a part of the complete data). By moving database shards on different MySQL instances, PolarDB-X implements database scale-out and improves the overall access to and the storage capacity of PolarDB-X databases.

PolarDB-X provides sharding rules, allowing you to select a partitioning policy that fits your business data characteristics. This ensures low latency for online database operations for transactions in high-concurrency scenarios. Therefore, when you use PolarDB-X, choosing the shard key is one of the important steps in database table structure design. The general principles are as follows:

- PolarDB-X performs well when writing data at the frontend. Most operations of such businesses are performed based on a specific database entity. For example, the business operations of the Internet are performed for users, the business operations of Internet of Things (IoT) are performed for devices and vehicles, the business operations of banks and government agencies are performed for customers, and the business operations of e-commerce independent software vendors (ISVs) and catering ISVs are performed for merchants. The data of such businesses can be partitioned by database entity. This, combined with global secondary indexes and eventually consistent transactions, can address the requirements on databases for large data volume, high concurrency, and low latency.
- For backend businesses, a batch of data is filtered and displayed on pages by condition and then processed and written back to the database. This is a business scenario in which PolarDB-X can partially address the needs. In this case, a large number of single-table associations and multi-table associations may exist, multiple filtering conditions are combined for DELETE and SELECT operations, and a large number of multi-table transactions are processed. Data partitioning by entity is recommended for such scenarios. If database processing is tightly related to time, data can be partitioned by time.

The following figure shows how data partitioning works.

Figure of data partitioning



ID	name	age
1	kobe	23
3	james	2
4	jordan	35
5	jordan	4
9	james	25
11	kobe	33
20	james	18

ID	name	INFO
1	kobe	23
4	jordan	35
5	jordan	4
11	kobe	33

ID	name	INFO
3	james	2
9	james	25
20	james	18

9.4.2. Smooth scale-out

To scale out a PolarDB-X instance, you can add ApsaraDB RDS for MySQL instances and migrate the original database shards to the new ApsaraDB RDS for MySQL instances.

Smooth scale-out is an online horizontal expansion method. It smoothly migrates the original database shards to the new ApsaraDB RDS for MySQL instances and increases the overall data storage capacity by adding ApsaraDB RDS for MySQL instances, which reduces the pressure on each RDS instance to process data.

How PolarDB-X scale-our works

Follow these steps:

1. Create a scale-out plan.

   Select a new ApsaraDB RDS for MySQL instance and database shards to be migrated. After the task is submitted, the system automatically creates a database and an account on the destination instance and submits a task for data migration and synchronization.

2. Perform full data migration.

   The system selects a time point before the current time and copies and migrates all data generated before this time point.

3. Perform incremental synchronization.

   After a full migration is completed, incremental data is synchronized according to the incremental change logs generated between a time point before the full migration and the current time, and eventually, the data is synchronized from the source database shard to the destination database shard in real time.

4. Verify data.

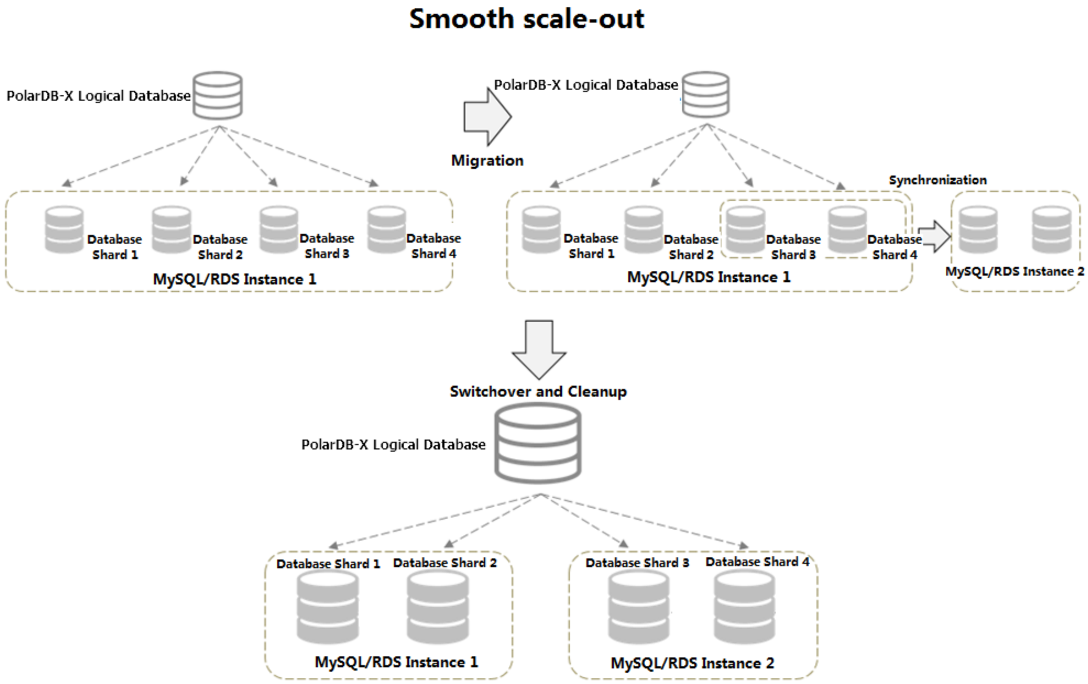
   When the incremental data is synchronized in quasi-real time, the system automatically performs full data verification and corrects inconsistent data caused by synchronization latency.

5. Disable the application service and switch routes.

After verification, the incremental data is still synchronized in quasi-real time, and a specified time is selected for the switch. To ensure strict data consistency, we recommend that you disable the service (you can also not disable the service but the same data may be overwritten at a high concurrency). The engine layer switches routes based on database sharding rules to switch subsequent traffic to the new database. The switching process can be completed within seconds.

The following figure shows data migration between database shards.

Scale-out



To ensure data security and facilitate rollback of a scale-out task, data synchronization continues after the routing rule is switched. After the data O&M personnel confirm that the service is normal, you can clean up data in the source database shard in the console.

The whole scale-out process has little impact on services of the upper layer (some services may be affected if the instance type of the ApsaraDB RDS for MySQL instance is not satisfactory or its traffic pressure is high). If the service is not disabled during the switch, we recommend that you perform this operation when the database access traffic is low to reduce the possibility of concurrently updating the same data.

9.4.3. Read/write splitting

The read/write splitting function of PolarDB-X is a relatively transparent policy to switch over the read traffic for ApsaraDB RDS for MySQL instances.

You can add read-only ApsaraDB RDS for MySQL instances and adjust their read weights in the PolarDB-X console without code modification if your business applications can tolerate the latency of data synchronization between read-only instances and the primary instance. The read traffic is proportionally adjusted between the primary ApsaraDB RDS for MySQL instance and multiple read-only ApsaraDB RDS for MySQL instances. Write operations and transaction operations are performed on the primary ApsaraDB RDS for MySQL instance.

Note that a latency exists for data synchronization between the primary instance and read-only instances. When a large data definition language (DDL) statement is executed or a large volume of data is being corrected, the latency may be over one minute. Therefore, consider whether your business can tolerate the impact before using this function.

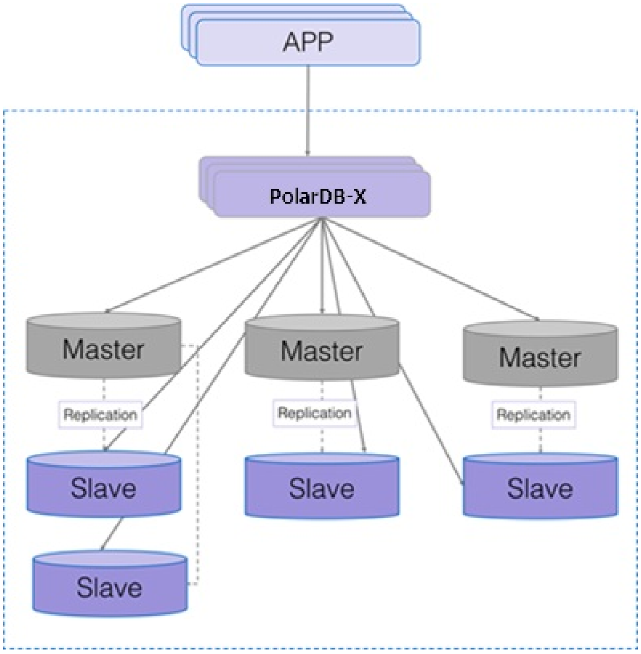
Adding read-only instances improves the read performance linearly. For example, if there is only one read-only instance, the read performance is doubled after one other read-only instance is added or tripled after two other read-only instances are added.

Traffic distribution and instance addition for read/write splitting

The read/write splitting function of PolarDB-X requires no modification of application code. You only need to add read-only instances and adjust the weights of read operations in the PolarDB-X console, to proportionally adjust the read traffic between the primary instance and multiple read-only instances. The write operations are performed on the primary instance.

Adding read-only instances improves the read performance linearly. For example, if there is one read-only instance, the read performance is doubled after one other read-only instance is added or tripled after two other read-only instances are added, as shown in the following figure.

Traffic distribution and expansion for read/write splitting

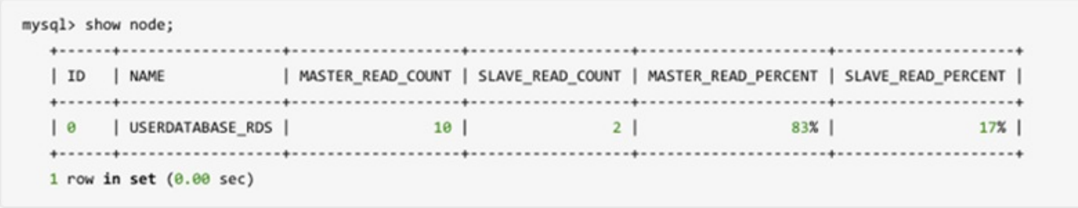


All data in the read operations on a read-only instance is asynchronously synchronized from the primary instance with a millisecond-level latency. For SQL statements that require high real-time performance, you can specify the primary instance through PolarDB-X Hint to execute these SQL statements, as shown in the following code:

```
/*TDDL:MASTER/select * from tddl5_users;
```

PolarDB-X allows you to run SHOW NODE to view the actual distribution of read traffic, as shown in the following figure.

SHOW NODE to view the actual distribution of read traffic



```
mysql> show node;
  +------+------------------+-------------------+------------------+---------------------+--------------------+
  | ID   | NAME             | MASTER_READ_COUNT | SLAVE_READ_COUNT | MASTER_READ_PERCENT | SLAVE_READ_PERCENT |
  +------+------------------+-------------------+------------------+---------------------+--------------------+
  | 0    | USERDATABASE_RDS |                10 |                2 |                 83% |                17% |
  +------+------------------+-------------------+------------------+---------------------+--------------------+
  1 row in set (0.00 sec)
```

Read/write splitting in non-partition mode

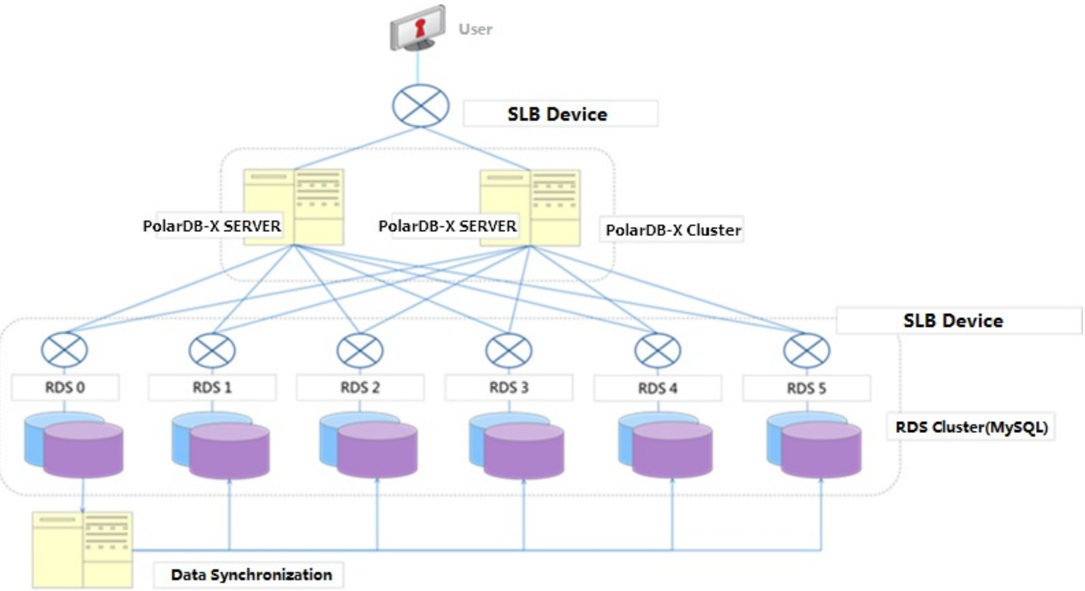
The read/write splitting function of PolarDB-X can be used independently in non-partition mode.

When you select an ApsaraDB RDS for MySQL instance for creating a PolarDB-X database in the PolarDB-X console, you can directly introduce a logical database on the ApsaraDB RDS for MySQL instance to the PolarDB-X database for read/write splitting without data migration.

9.4.4. Service upgrade and downgrade

A PolarDB-X instance consists of multiple server nodes that are deployed in a cluster and provides services externally through Server Load Balancer (SLB) and Domain Name System (DNS). The server nodes of PolarDB-X do not synchronize states, they process external requests in a balanced manner. When the processing capability of the server cluster is insufficient, server nodes can be added in real time to improve the service capability. If the resource utilization of all PolarDB-X server nodes in the cluster is low, you can remove some server nodes to downsize the cluster, and lower the capability of the service layer, for the elastic scaling of service capabilities. Figure of service upgrade and downgrade shows the details.

Figure of service upgrade and downgrade



9.4.5. Account and permission system

The account and permission system of PolarDB-X is used in the same way as MySQL, but does not support authorization across multiple databases, and has fewer permissions than MySQL. The system supports statements including GRANT, REVOKE, SHOW GRANTS, CREATE USER, DROP USER, and SET PASSWORD. It can be used to grant database-level and table-level permissions, but global and column-level permissions are not supported.

Account rules:

- The administrator account created in the console has all permissions.
- Only the administrator account can create and authorize accounts. Other accounts can only be created and authorized 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.

Currently, eight table-associated basic permissions are supported: CREATE, DROP, ALTER, INDEX, INSERT, DELETE, UPDATE, and SELECT. Among these operations:

- The TRUNCATE operation requires the table-level DROP permission.
- The REPLACE operation requires the table-level INSERT and DELETE permissions.
- The CREATE INDEX and DROP INDEX operations require the table-level INDEX permission.
- The CREATE SEQUENCE operation requires the database-level CREATE permission.
- The DROP SEQUENCE operation requires the database-level DROP permission.
- The ALTER SEQUENCE operation requires the database-level ALTER permission.
- The INSERT ON DUPLICATE UPDATE statement requires the table-level INSERT and UPDATE permissions.

9.4.6. PolarDB-X sequence

The PolarDB-X sequence of PolarDB-X (a 64-digit number of the signed BIGINT type in MySQL) aims to generate a globally unique number sequence (not necessarily in increments). This sequence is usually used to generate keys such as a primary key column and a unique key.

The PolarDB-X sequence of PolarDB-X can be implicitly used. When a table is partitioned to table shards, the primary key is auto_increment, and business data is inserted with no primary key specified, the globally unique primary key is automatically set, just like the single-instance MySQL database.

The PolarDB-X sequences can also be explicitly used. You can run `select xxx_seq.nextval from dual where count=?` to obtain one or more PolarDB-X sequences for other use in the application.

9.4.7. Second-level monitoring

PolarDB-X allows users to run **SHOW FULL STATS** to implement second-level monitoring. This command operates with the business monitoring system of PolarDB-X or third-party open-source monitoring software to provide better monitoring and alerting effect.

The following table describes the metrics supported by this command.

Metric	Description
QPS	The logical queries per second (QPS) specifying queries from an application to a PolarDB-X instance.
RDS_QPS	The QPS from a PolarDB-X instance to an ApsaraDB RDS for MySQL instance is called physical QPS.

Metric	Description
ERROR_PER_SECOND	The number of errors per second, which is the sum of SQL syntax errors, primary key conflicts, system errors, and connectivity errors.
VIOLATION_PER_SECOND	The number of primary key conflicts or unique key conflicts per second.
MERGE_QUERY_PER_SECOND	The number of queries merged from multiple table shards.
ACTIVE_CONNECTIONS	The number of connections in use.
CONNECTION_CREATE_PER_SECOND	The number of connections created per second.
RT(MS)	The logical response time (RT) for a response from an application to a PolarDB-X instance.
RDS_RT(MS)	The physical RT for a response from a PolarDB-X instance to an ApsaraDB RDS for MySQL instance.
NET_IN(KB/S)	The network traffic received by PolarDB-X
NET_OUT(KB/S)	The network traffic sent by PolarDB-X
THREAD_RUNNING	The number of running threads.
HINT_USED_PER_SECOND	The number of queries with hints per second.
HINT_USED_COUNT	The total number of queries with hints since startup.
AGGREGATE_QUERY_PER_SECOND	The number of aggregate queries per second.
AGGREGATE_QUERY_COUNT	The total number of historical aggregate queries.
TEMP_TABLE_CREATE_PER_SECOND	The number of temporary tables created per second.
TEMP_TABLE_CREATE_COUNT	The total number of temporary tables created since startup.
MULTI_DB_JOIN_PER_SECOND	The number of cross-database JOIN queries per second.
MULTI_DB_JOIN_COUNT	The total number of cross-database JOIN queries since startup.

9.4.8. Distributed SQL engine

The distributed SQL engine of PolarDB-X is designed to achieve high compatibility with a single-instance MySQL database, to implement SQL push-down. PolarDB-X allows you to perform SQL operations, such as analysis, optimization, routing, and data aggregation.

The core principles of SQL push-down are as follows:

- Process data as close to the data as possible.
- Reduce data transfers over the network.
- Reduce data processing on PolarDB-X and offload the processing work to the lower-level data nodes whenever possible.
- Make full use of the features and capabilities of database storage.

9.4.9. High-availability architecture

Automatic traffic switchover of PolarDB-X Server

The PolarDB-X Server component of a PolarDB-X instance consists of multiple server nodes and provides services as a single connection through a load balancing service. When a PolarDB-X server fails, its traffic switches over to another PolarDB-X server in seconds. The entire failover process is transparent to users, with no need to change the application code or restart the application.

Automatic traffic switchover

PolarDB-X supports the read/write splitting function. You can sign in to the console, choose **DRDS Database > Read/Write Splitting**, and configure the function. This function allocates some read traffic to the secondary instances. PolarDB-X identifies the read SQL requests and distributes them to the primary and secondary ApsaraDB RDS for MySQL instances based on the configured ratio, to implement read/write splitting. A secondary instance allocated with only read traffic is called a read-only instance.

If multiple read-only instances are configured but one of them fails (the connection fails), PolarDB-X automatically withdraws the read traffic from the failed instance, and then re-allocates the traffic based on the ratio of read traffic in the remaining normal read-only instances.

The automatic traffic switchover process of read-only instances is transparent to users. You do not need to restart applications. When no read-only instance is available, read requests are still allocated proportionally to both the read-only and the primary instances, to prevent the primary instance from being overloaded. Errors are reported for the read requests allocated to read-only instances .

> ? **Note** All write requests and transactions are automatically routed to the primary instance for execution, regardless of the availability of read-only instances.

9.4.10. Software upgrade

- PolarDB-X automatically provides new versions of installed database software.
- Software upgrade is optional. It is carried out only upon your request.
- If PolarDB-X determines that your version has major security risks, it will notify you to schedule the upgrade. The PolarDB-X team will provide support during the entire upgrade process.
- The PolarDB-X upgrade process is generally completed within 5 minutes. During the upgrade process, there may be several transient database disconnections. There is minimal interruption to applications if the database reconnection (or connection pool) is properly configured for applications.

9.4.11. SQL compatibility

PolarDB-X is compatible with the MySQL protocols and supports most MySQL query syntax, common data manipulation language (DML) syntax, and data definition language (DDL) syntax. However, the pronounced architectural differences between distributed databases and single-instance databases restrict the usage of SQL. The compatibility and SQL restrictions are described as follows.

> **Note** Since there are many MySQL versions and the MySQL syntax and PolarDB-X versions are constantly updating, the compatibility discussed in this document is for reference only. Determine whether the selected MySQL version matches your business according to the actual test results.

PolarDB-X SQL restrictions

SQL restrictions are as follows:

- Custom data types and functions are not supported at present.
- Views, stored procedures, triggers, and cursors are not supported at present.
- Compound statements such as `BEGIN...END, LOOP...END LOOP, REPEAT...UNTIL...END REPEAT, and WHILE...DO...END WHILE` are not supported at present.
- Process control statements such as IF and WHILE are not supported at present.

Small syntax restrictions

DDL:

- CREATE TABLE tbl_name LIKE old_tbl_name does not support table sharding.
- CREATE TABLE tbl_name SELECT statement does not support table sharding.

DML:

- SELECT INTO OUTFILE, SELECT INTO DUMPFILE, and SELECTvar_name are not supported at present.
- INSERT DELAYED is not supported at present.
- Subqueries irrelevant to the WHERE condition are not supported at present.
- SQL subqueries that contain aggregation conditions are not supported at present.
- Variable references and operations in SQL statements are not supported at present, for example, `SET @c=1, @d=@c+1; SELECT @c, @d`.

Database management:

- SHOW WARNINGS does not support the LIMIT /COUNT combination.
- SHOW ERRORS does not support the LIMIT /COUNT combination.

Compatibility of PolarDB-X with SQL

Compatibility with MySQL protocols

PolarDB-X supports mainstream clients such as MySQL Workbench, Navicat For MySQL, and SQLyog.

> **Note** PolarDB-X supports the add, delete, modify, and query operations on databases. However, other special functions (such as import and diagnosis) have not been thoroughly tested.

PolarDB-X is compatible with the following DDL statements:

- `CREATE TABLE`
- `CREATE INDEX`

- `DROP TABLE`
- `DROP INDEX`
- `ALTER TABLE`
- `TRUNCATE TABLE`

PolarDB-X is compatible with the following DML statements:

- INSERT
- REPLACE
- UPDATE
- DELETE
- Subquery
- Scalar subquery
- Comparisons subquery
- Subquery with ANY, IN, or SOME
- Subquery with ALL
- Subquery by column
- Subquery with EXISTS or NOT EXISTS
- Subquery in the FROM clause
- SELECT

PolarDB-X is compatible with the following PREPARE statements:

- PREPARE
- EXECUTE
- DEALLOCATE PREPARE

PolarDB-X is compatible with the following database management statements

- SET
- SHOW
- KILL 'PROCESS_ID' (PolarDB-X only supports the KILL 'PROCESS_ID' command but does not support the KILL QUERY command.)
- SHOW COLUMNS
- SHOW CREATE TABLE
- SHOW INDEX
- SHOW TABLES
- SHOW TABLE STATUS
- SHOW TABLES
- SHOW VARIABLES
- SHOW WARNINGS
- SHOW ERRORS

> **Notice** Other SHOW commands are delivered to the database for processing by default, and the returned result data in different shards are not merged.

PolarDB-X is compatible with the following database tool statements:

- DESCRIBE
- EXPLAIN
- USE

Custom instructions of PolarDB-X are as follows:

- SHOW SEQUENCES, CREATE SEQUENCE, ALTER SEQUENCE, and DROP
- SEQUENCE. It manages PolarDB-X sequences.
- SHOW PARTITIONS FROM TABLE. It queries table shard keys.
- SHOW TOPOLOGY FROM TABLE. It queries the physical topology of a table.
- SHOW BROADCASTS. It queries all broadcast tables.
- SHOW RULE [FROM TABLE]. It queries the table sharding rule.
- SHOW DATASOURCES. It queries data sources of the backend database connection pool.
- SHOW DBLOCK/RELEASE DBLOCK. It defines the distributed LOCK.
- SHOW NODE. It queries the database read and write traffic.
- SHOW SLOW. It queries the slow SQL statements.
- SHOW PHYSICAL_SLOW. It queries slow SQL statements executed in the physical database.
- TRACE SQL_STATEMENT/SHOW TRACE. It traces the SQL statement execution process.
- EXPLAIN [DETAIL/EXECUTE] SQL_STATEMENT. It analyzes the SQL execution plans of PolarDB-X and physical databases.
- RELOAD USERS. It synchronizes the user information from PolarDB-X Console to the PolarDB-X Server.
- RELOAD SCHEMA. It clears data caches in the corresponding PolarDB-X database, such as cache of SQL parsing, syntax tree, and table structure.
- RELOAD DATASOURCES. It rebuilds a connection pool that connects the backend to all databases.

Database functions:

- SQL statements with shard keys are supported by all MySQL functions.
- SQL statements without a shard key are supported by only some functions.
- Operator functions

Function	Description
AND, &&	Logical AND
=	Assigns a value (a part of the SET statement or a part of the SET clause in the UPDATE statement).
BETWEEN... AND...	Determines a certain range of a value.
BINARY	Converts a string into a binary string.
&	Bitwise AND
~	Bitwise negation
^	Bitwise Exclusive OR (XOR)

Function	Description
DIV	Returns an integer obtained from integer division.
/	Division operator
<=>	NULL-safe equal operator
=	Equal operator, it compares the equality of two strings
>=	Operator, greater than or equal to
>	Greater than
IS NOT NULL	Tests for a non-NULL value.
IS NOT	Tests for a non-Boolean value.
IS NULL	It tests for a NULL value.
IS	Tests for a Boolean value.
<<	Bitwise left shift operator
<=	Operator, less than or equal to
<	Operator, less than
LIKE	Compares a character string to a specified string pattern.
-	Minus operator
%,	Returns the remainder of a number divided by another number.
NOT BETWEEN... AND...	Determines a certain range that a value is not in.
! =, <>	Operator, not equal to
NOT LIKE	Finds a specific character string that does not match a specified pattern.
NOT REGEXP	NOT operator in regular expressions
NOT, !	NOT
OR	Logical OR
+	Plus operator
REGEXP	Uses a regular expression for matching.
>>	Bitwise right shift operator
RLIKE	Uses a regular expression for matching. It is the same as REGEXP。

Function	Description
*	Multiplication operator
-	Takes the opposite value of the parameter.
XOR	Logical XOR
Coalesce	Returns the first non-NULL parameter.
GREATEST	Returns the largest parameter value.
LEAST	It returns the smallest parameter value.
STRCMP	Compares two strings.

- Process control functions

Function	Description
CASE	Case operator
IF()	If/else structure
IFNULL()	Null if/else structure
NULLIF()	If expr1 = expr2, NULL is returned.

- Numeric functions

Function	Description
ABS()	Returns the absolute value.
ACOS()	Returns the arc cosine of a number.
ASIN()	Returns the arc sine of a number.
ATAN2()	Returns the arc tangent of two parameters.
ATAN()	Returns the arc tangent of a parameter.
CEIL()	Obtains the smallest integer greater than or equal to a number.
CEILIG()	Obtains the smallest integer greater than or equal to a number.
CONV()	Converts a number between different number bases.
COS()	Returns the cosine of a number.
COT()	Returns the cotangent of a number.
CRC32()	Calculates the cyclic redundancy check (CRC) value.

Function	Description
DEGREES()	Converts a radian to a degree.
DIV	Returns an integer obtained from integer division.
EXP()	Returns e raised to the power of the specified number.
FLOOR()	Obtains the largest integer less than or equal to a number.
LN()	Returns the natural logarithm of a parameter.
LOG10()	Returns the logarithm with the base 10 of the parameter.
LOG2()	Returns the logarithm with the base 2 of the parameter.
LOG()	Returns the natural logarithm of the first parameter.
MOD()	Returns the remainder of a number.
%,MOD	Returns the remainder of a number divided by another number.
PI()	Returns the value of Pi.
POW()	Returns N power of the first parameter, where N is the second parameter.
POWER()	Returns N power of the first parameter, where N is the second parameter.
RADIANS()	Converts a parameter into a radian.
RAND()	Returns a random floating-point number.
ROUND()	Rounds up or down to an integer.
SIGN()	Returns the positive or negative sign of a parameter.
SIN()	Returns the sine value of a parameter.
SQRT()	Returns the square root of a parameter.
TAN()	Returns the tangent of a parameter value.
TRUNCATE(	Truncates to the specified decimal place.

- String functions

Function	Description
ASCII()	Returns the ASCII value of a character.
BIN()	Returns the binary value of a character.
BIT_LENGTH()	Returns the bit length of a string.

Function	Description
CHAR_LENGTH()	Returns the number of characters in a string.
CHAR()	Converts an input integer into a character.
CHARACTER_LENGTH()	Returns the number of characters in a string. It is the same as CHAR_LENGTH().
CONCAT_WS()	Connects the input parameters by using the specified separator.
CONCAT()	Returns a connection string.
ELT()	Returns the string at the index number.
EXPORT_SET()	-
FIELD()	Returns the index position of the first parameter in subsequent parameters.
FIND_IN_SET()	Returns the index position of the first parameter in the second parameter.
FORMAT()	Returns the formatted numbers of the specified decimal places.
HEX()	It converts a decimal number or string into a hexadecimal number.
INSERT()	Inserts a substring of a specified number of characters at the specified place.
INSTR()	Returns the index position where the substring appears for the first time.
LCASE()	Converts to lowercase letters. It is the same as LOWER().
LEFT()	Returns the characters of the specified number that is the furthest left.
LENGTH()	Returns the number of bytes of a string.
LIKE	Finds a specific character string matches a specified pattern.
LOCATE()	Returns the position where the substring appears for the first time.
LOWER()	Converts to lowercase letters.
LPAD()	Pads the left side of a string with a specific set of characters.
LTRIM()	Removes spaces at the beginning.
MAKE_SET()	Returns a set value (a string containing substrings separated by , characters) consisting of the characters specified in the first argument.
MID()	Extracts a substring from a string (starting at the specified position).

Function	Description
NOT LIKE	Finds a specific character string that does not match a specified pattern.
NOT REGEXP	Performs a pattern match of a string expression against a pattern.
OCT()	Converts a number to an octal number by a string.
OCTET_LENGTH()	Returns the number of bytes of a string. It is the same as LENGTH().
ORD()	Returns the code for the leftmost character of the given parameter.
POSITION()	Returns the position where the sub-string occurs for the first time. It is the same as LOCATE().
QUOTE()	Escapes parameters for use in SQL statements.
REPEAT()	Repeats a string for a specified number of times.
REPLACE()	Replaces the specified string in all the places where it appears.
REVERSE()	Reverses characters in a string.
RIGHT()	Returns the characters of the specified number that is the furthest right.
RPAD()	Pads strings for the specified number of times from the right.
RTRIM()	Removes spaces at the end.
SPACE()	Returns a string consisting of specified spaces.
STRCMP()	Compares two strings.
SUBSTR()	Returns the specified substring.
SUBSTRING_INDEX()	Returns the substring that appears for a specified number of times and in front of a separator in a string.
SUBSTRING()	Returns the specified substring.
TRIM()	Removes spaces at the beginning and end.
UCASE()	Converts a string to all uppercase. It is the same as UPPER().
UNHEX()	Returns a string that is the hexadecimal value of the parameter.
UPPER()	Converts a string to uppercase.

- Time functions

Function	Description
ADDDATE()	Adds a time value (an interval) to a date.
ADDTIME()	Adds a time interval to a time/datetime and then returns the time/datetime.
CURDATE()	Returns the current date.
CURRENT_DATE()	Returns the current date. It is the same as CURDATE().
CURRENT_TIME()	Returns the current time. It is the same as CURTIME().
CURRENT_TIMESTAMP()	Returns the current date and time. It is the same as NOW().
CURTIME()	Returns the current time.
DATE_ADD()	Adds a time value (an interval) to a date.
DATE_FORMAT()	Formats the date as required.
DATE_SUB()	Subtracts a specified time value (an interval) from a date.
DATE()	Extracts the date from the date expression or datetime expression.
DATEDIFF()	Subtracts one date from the other date.
DAY()	Returns the day (0-31) of a month for the specified date. It is the same as DAYOFMONTH().
DAYNAME()	Returns the weekday for a date.
DAYOFMONTH()	Returns the day (0-31) of a month for the specified date.
DAYOFWEEK()	Returns the day of a week (1 for Sunday and 7 for Saturday) for a date.
DAYOFYEAR()	Returns the day (1-366) of a year for a date.
EXTRACT()	Extracts a part of a date.
FROM_DAYS()	Converts a day to a date.
FROM_UNIXTIME()	Formats a UNIX timestamp as a date.
GET_FORMAT()	Returns a string of the date format.
HOUR()	Extracts hours from input time parameters.
LAST_DAY()	Returns the last day of the month for the parameter.
LOCALTIME()	Returns the current date and time. It is the same as NOW().
LOCALTIMESTAMP, LOCALTIMESTAMP()	Returns the current date and time. It is the same same as NOW().

Function	Description
MAKEDATE()	Returns the date, containing the year and the number of days.
MAKETIME()	Constructs a time containing the hour, minute, and second.
MICROSECOND()	Returns the microsecond of a parameter.
MINUTE()	Returns the minute of a parameter.
MONTH()	Returns the month of the input date.
MONTHNAME()	Returns the name of a month.
NOW()	Returns the current date and time.
PERIOD_ADD()	Adds a period to a date containing the year and month.
PERIOD_DIFF()	Returns the number of months between two periods.
QUARTER()	Returns the quarter of the date parameter.
SEC_TO_TIME()	Converts the second into the time in 'HH:MM:SS' format.
SECOND()	Returns the second (0-59) of a minute.
STR_TO_DATE()	Converts the string to a date.
SUBDATE()	Subtracts a specified time value (an interval) from a date when three parameters are called. It is the same as DATE_SUB().
SUBTIME()	Subtracts a time interval from a time/datetime and then returns the time/datetime.
SYSDATE()	Returns the function execution time.
TIME_FORMAT()	Formats the time.
TIME_TO_SEC()	Converts a parameter into a second.
TIME()	Extracts the time of an input parameter.
TIMEDIFF()	Returns the difference between two time/datetime expressions.
TIMESTAMP()	Returns a datetime value or expression based on a date or datetime value. If there are two arguments specified with this function, it first adds the second argument to the first, and then returns a datetime value.
TIMESTAMPADD()	Adds a time interval to the datetime expression.
TIMESTAMPDIFF()	Subtracts a time interval from the datetime expression.
UNIX_TIMESTAMP()	Returns the UNIX timestamp.

Function	Description
UTC_DATE()	Returns the current UTC date.
UTC_TIME()	Returns the current UTC time.
UTC_TIMESTAMP()	Returns the current UTC date and time.
WEEKDAY()	Returns the weekly index, where 1 indicates Sunday and 7 indicates Saturday.
WEEKOFYEAR()	Returns the number of the week on the calendar for a date.
YEAR()	It returns the year.

- Type conversion functions

Function	Description
BINARY	Converts a string into a binary string.
CAST()	Converts a value into a type.
CONVERT()	It converts a value into a type.

9.4.12. Table sharding

PolarDB-X provides convenient table sharding and changing functions, allowing you to flexibly partition a table into table shards, to glue table shards to a table, and to transfer data from one table shard to another.

9.4.13. Multi-zone instances

PolarDB-X allows you to select a multi-zone PolarDB-X instance. This ensures the PolarDB-X instance availability when one of the zones is unavailable.

9.4.14. Zone-disaster recovery

PolarDB-X provides the zone-disaster recovery function, supporting migration between single-zone instances and dual-zone instances. Zone-based disaster recovery can be performed if an inappropriate zone is selected for the target PolarDB-X instance or the available ApsaraDB RDS for MySQL instances in the target PolarDB-X zone are insufficient.

10.AnalyticDB for PostgreSQL

10.1. What is AnalyticDB for PostgreSQL?

AnalyticDB for PostgreSQL is a distributed analytic database service that leverages the massively parallel processing (MPP) architecture, where each instance is composed of multiple compute nodes. AnalyticDB for PostgreSQL provides MPP warehousing services that support horizontal scaling of storage and compute capabilities, online analysis of petabytes of data, and offline processing of Extract, Transform, and Load (ETL) tasks.

AnalyticDB for PostgreSQL is developed based on the PostgreSQL kernel and has the following features:

- Supports the SQL:2003 standard, OLAP aggregate functions, views, Procedural Language for SQL (PL/SQL), user-defined functions (UDFs), and triggers. AnalyticDB for PostgreSQL is partially compatible with the Oracle syntax.
- Supports horizontal scaling of storage and compute capabilities based on the MPP architecture. AnalyticDB for PostgreSQL also supports range and list partitioning.
- Supports row store, column store, and multiple indexes. AnalyticDB for PostgreSQL also supports multiple compression methods based on column store to reduce storage costs.
- Supports standard database isolation levels and distributed transactions to ensure data consistency.
- Provides the vector computing engine and the CASCADE-based SQL query optimizer to ensure high-performance SQL analysis.
- Uses a primary/secondary architecture to ensure dual-copy data storage and service availability.
- Provides online scaling, system monitoring, and disaster recovery to reduce O&M costs.

10.1.1. Scenarios

This topic describes the OLAP data analysis services that AnalyticDB for PostgreSQL supports.

- Extract, Transform, and Load (ETL) for offline data processing

  AnalyticDB for PostgreSQL provides the following benefits that make it ideal to optimize complex SQL queries as well as aggregate and analyze large amounts of data:

  - Supports standard SQL syntax, OLAP window functions, and stored procedures.
  - Provides the CASCADE-based SQL query optimizer to enable complex queries without the need for tuning.
  - Built on the MPP architecture that supports horizontal scaling of storage and compute capabilities to analyze and process petabytes of data.
  - Provides column store-based high-performance aggregation of large tables at a high compression ratio to maximize storage capacity.

- Online high-performance query

  AnalyticDB for PostgreSQL provides the following benefits for real-time exploration, warehousing, and updating of data:

  - Allows you to write and update high-throughput data by performing INSERT, UPDATE, and DELETE operations.

  - Allows you to query data based on row store and multiple indexes to obtain results within milliseconds. These indexes include B-tree indexes, bitmap indexes, and hash indexes.
  - Supports distributed transactions, standard database isolation levels, and HTAP.

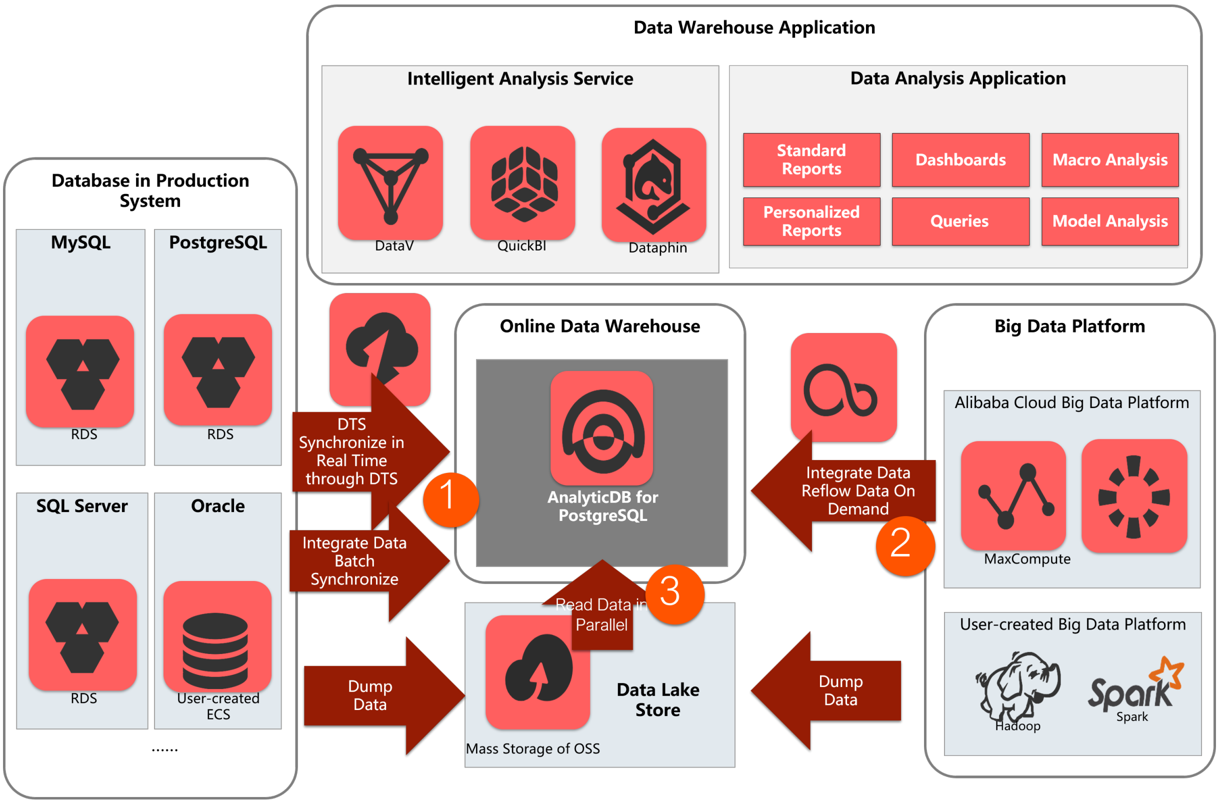
- Multi-model data analysis

  AnalyticDB for PostgreSQL provides the following benefits for processing unstructured data from a variety of sources:

  - Supports the PostGIS extension for geographic data analysis and processing.
  - Uses the MADlib library of in-database machine learning algorithms to implement AI-native databases.
  - Provides high-performance retrieval and analysis of unstructured data such as images, speech, and text by means of vector retrieval.
  - Supports formats such as JSON and can process and analyze semi-structured data such as logs.

Typical scenarios

AnalyticDB for PostgreSQL is applicable to the following scenarios:


Data Warehouse Application
Intelligent Analysis Service
DataV
QuickBI
Dataphin
Data Analysis Application
Standard Reports
Dashboards
Macro Analysis
Personalized Reports
Queries
Model Analysis
Database in Production System
MySQL
RDS
PostgreSQL
RDS
SQL Server
RDS
Oracle
User-created ECS
......
DTS Synchronize in Real Time through DTS
1
Integrate Data Batch Synchronize
Dump Data
Online Data Warehouse
AnalyticDB for PostgreSQL
Read Data in Parallel
3
Data Lake Store
Mass Storage of OSS
Integrate Data Reflow Data On Demand
2
Dump Data
Big Data Platform
Alibaba Cloud Big Data Platform
MaxCompute
User-created Big Data Platform
Hadoop
Spark

- Data warehousing service

  Data Transmission Service (DTS) can synchronize data in real time in production system databases such as ApsaraDB RDS for MySQL, ApsaraDB RDS for PostgreSQL, and PolarDB as well as traditional databases such as Oracle and SQL Server. Data can also be batch synchronized to AnalyticDB for PostgreSQL by using Data Integration. AnalyticDB for PostgreSQL supports ETL operations on large amounts of data. You can also use DataWorks to schedule these tasks. AnalyticDB for PostgreSQL also provides high-performance online analysis capabilities and can use Quick BI, DataV, Tableau, and FineReport for report presentation and real-time query.

- Big data analytics platform

You can use Data Integration or OSS to import large amounts of data from MaxCompute, Hadoop, and Spark to AnalyticDB for PostgreSQL for high-performance analysis, processing, and exploration.

- Data lake analytics

  AnalyticDB for PostgreSQL can use foreign tables to access the large amounts of data stored in OSS in parallel and build an Alibaba Cloud data lake analytics platform.

10.2. Benefits

This topic describes the benefits of AnalyticDB for PostgreSQL.

- Real-time analysis

  Built on the MPP architecture that supports horizontal scaling and can respond to queries on petabytes of data within seconds. AnalyticDB for PostgreSQL supports the leading vector computing feature and intelligent indexes of column store. It also supports the CASCADE-based SQL query optimizer to enable complex queries without the need for tuning.

- Stability and reliability

  Provides ACID properties for distributed transactions. Transactions are consistent across nodes and all data is synchronized between primary and secondary nodes. AnalyticDB for PostgreSQL supports distributed deployment and provides transparent monitoring, switching, and restoration to secure your data infrastructure.

- Easy to use

  Supports rich SQL syntax and functions, Oracle functions, stored procedures, user-defined functions (UDFs), and isolation levels of transactions and databases. You can use popular business intelligence (BI) software and ETL tools online.

- Ultra-high performance

  Supports row store, column store, and multiple indexes. The vector engine provides high-performance analysis and computing capabilities. The CASCADE-based SQL query optimizer enables complex queries without the need for tuning. AnalyticDB for PostgreSQL supports high-performance parallel import of data from OSS.

- Flexible scalability

  Enables you to scale out compute nodes as well as CPU, memory, and storage resources on demand to improve OLAP performance.

  Supports transparent OSS operations. OSS offers a larger storage capacity for cold data that does not require online analysis.

  Supports online scaling to add, remove, modify, and query data during data redistribution.

- Resource isolation

  Supports multi-tenant parallel execution on a cluster by using multiple instances. Tasks from tenants are submitted to queues on different instances for execution. Resources of each AnalyticDB for PostgreSQL instance are isolated among tenants.

- Permission management

  Allows you to configure and manage tenants in a dynamic and centralized manner. You can also isolate resources and query usage statistics of resources. Management of multi-level tenants is supported.

- Resource scheduling

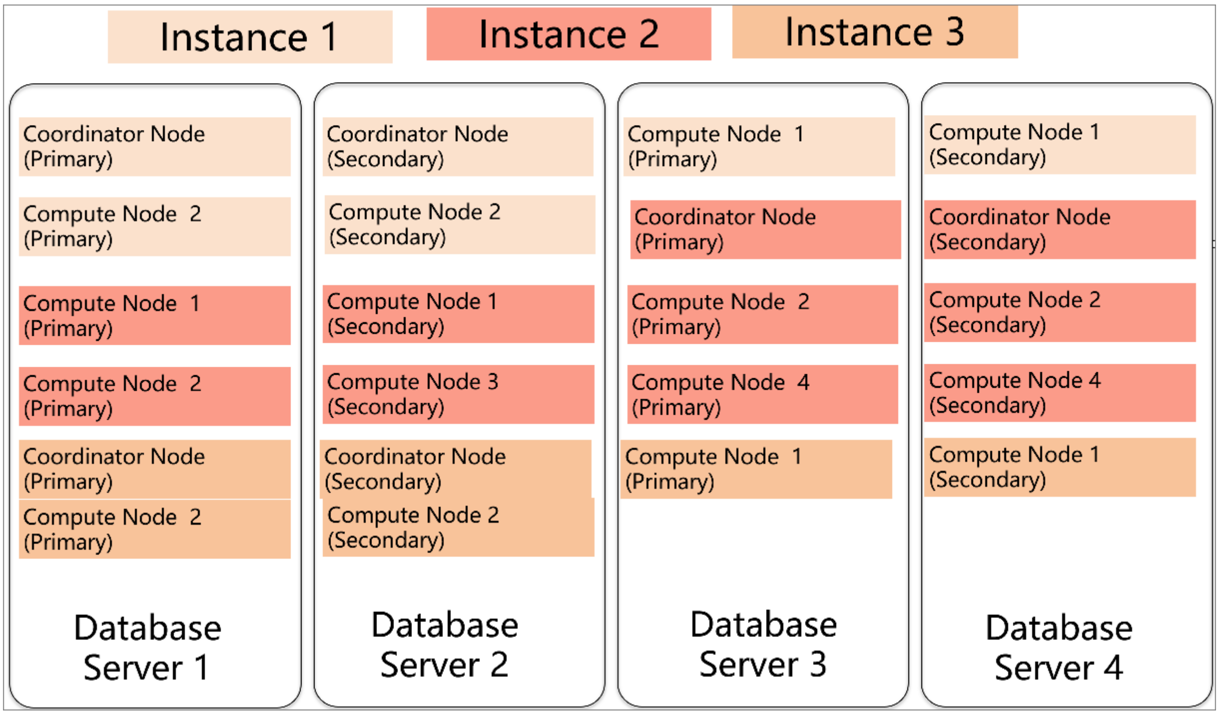
  Supports multi-tenant scheduling of multiple clusters and resource pools.

10.3. Architecture

This topic describes the physical architecture and logical architecture of an AnalyticDB for PostgreSQL cluster.

Physical architecture of a cluster

The following figure shows the physical architecture of an AnalyticDB for PostgreSQL cluster.

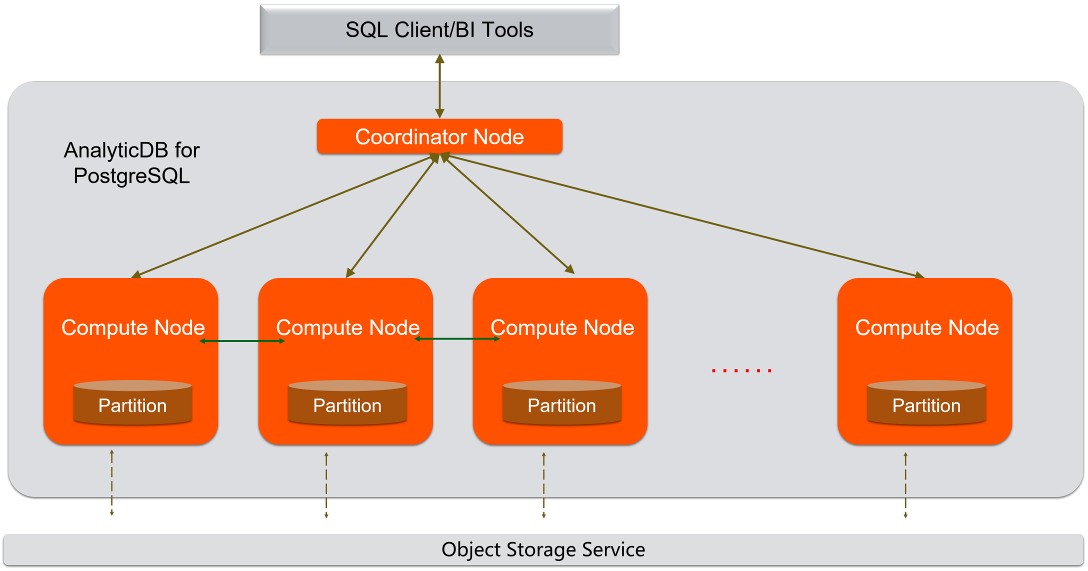

Instance 1
Instance 2
Instance 3
Coordinator Node (Primary)
Compute Node 2 (Primary)
Compute Node 1 (Primary)
Compute Node 2 (Primary)
Coordinator Node (Primary)
Compute Node 2 (Primary)
Database Server 1
Coordinator Node (Secondary)
Compute Node 2 (Secondary)
Compute Node 1 (Secondary)
Compute Node 3 (Secondary)
Coordinator Node (Secondary)
Compute Node 2 (Secondary)
Database Server 2
Compute Node 1 (Primary)
Coordinator Node (Primary)
Compute Node 2 (Primary)
Compute Node 4 (Primary)
Compute Node 1 (Primary)
Database Server 3
Compute Node 1 (Secondary)
Coordinator Node (Secondary)
Compute Node 2 (Secondary)
Compute Node 4 (Secondary)
Compute Node 1 (Secondary)
Database Server 4

You can create multiple AnalyticDB for PostgreSQL instances in a physical cluster of AnalyticDB for PostgreSQL by using the management and control system. Each instance consists of a coordinator node and multiple compute nodes.

- The coordinator node is used for access from applications. It receives connection requests and SQL query requests from clients and dispatches computing tasks to compute nodes. The cluster deploys a secondary node of the coordinator node on an independent physical server and replicates data from the primary node to the secondary node for failover. The secondary node does not accept external connections.
- Compute nodes are independent instances in AnalyticDB for PostgreSQL. Data is evenly distributed across compute nodes by hash value or RANDOM function, and is analyzed and computed in parallel. Each compute node uses a primary/secondary architecture for automatic failover.

Logical architecture of an instance

You can create multiple instances within an AnalyticDB for PostgreSQL cluster. The following figure shows the logical architecture of an instance.AnalyticDB for PostgreSQL



Data is distributed across compute nodes by hash value or RANDOM function of a specified distribution column. Each compute node uses a primary/secondary architecture to ensure dual-copy storage. High-performance network communication is supported across nodes. When the coordinator node receives a request from an application, the coordinator node parses and optimizes SQL statements to generate a distributed execution plan. After the coordinator node sends the execution plan to the compute nodes, the compute nodes perform massively parallel processing of the plan.

10.4. Features

This topic describes the features of AnalyticDB for PostgreSQL.

Distributed architecture

AnalyticDB for PostgreSQL is built on the massively parallel processing (MPP) architecture. Data is distributed evenly across nodes by hash value or RANDOM function, and is analyzed and computed in parallel. Storage and compute capabilities are scaled out by adding nodes to ensure a quick response as the data volume increases.

AnalyticDB for PostgreSQL supports distributed transactions to ensure data consistency among nodes. It supports three transaction isolation levels: SERIALIZABLE, READ COMMITTED, and READ UNCOMMITTED.

High-performance data analysis

AnalyticDB for PostgreSQL supports column store and row store for tables. Row store provides high update performance. Column store provides high OLAP aggregate analysis performance for tables. AnalyticDB for PostgreSQL supports B-tree indexes, bitmap indexes, and hash indexes to enable high-performance analysis, filtering, and query.

AnalyticDB for PostgreSQL uses the CASCADE-based SQL query optimizer. AnalyticDB for PostgreSQL combines the cost-based optimizer (CBO) with the rule-based optimizer (RBO) to provide SQL optimization features such as automatic subquery decorrelation. These features enable complex queries without the need for tuning.

High-availability service

AnalyticDB for PostgreSQL builds a system for automatic monitoring, diagnosis, and troubleshooting based on the Apsara system. This helps reduce O&M costs.

The coordinator node compiles and optimizes SQL statements by storing database metadata and receiving query requests from clients. The coordinator node uses a primary/secondary architecture to ensure strong consistency of metadata. If the primary coordinator node fails, the service is automatically switched to the secondary coordinator node.

Data synchronization methods and tools

You can use Data Transmission Service (DTS) or DataWorks Data Integration to synchronize data from MySQL or PostgreSQL databases to AnalyticDB for PostgreSQL. You can use popular Extract, Transform, and Load (ETL) tools to import ETL data to and schedule jobs in AnalyticDB for PostgreSQL databases. You can also use standard SQL syntax to query data from formatted files stored in OSS by using foreign tables in real time.AnalyticDB for PostgreSQL

AnalyticDB for PostgreSQL supports popular business intelligence (BI) reporting tools such as Quick BI, DataV, Tableau, and FineReport. It also supports ETL tools, including Informatica and Kettle.

Data security

AnalyticDB for PostgreSQL supports the configuration of whitelists. You can add up to 1,000 IP addresses of servers to a whitelist to allow access to your instance and control risks from access sources. AnalyticDB for PostgreSQL also supports Anti-DDoS to monitor inbound traffic in real time. When large amounts of malicious traffic is identified, the traffic is scrubbed by means of IP filtering. If traffic scrubbing is insufficient, blackhole filtering is triggered.

Supported SQL features

- Supports row store and column store.
- Supports multiple indexes, including B-tree indexes, bitmap indexes, and hash indexes.
- Supports distributed transactions and standard isolation levels to ensure data consistency among nodes.
- Supports character, date, and arithmetic functions.
- Supports stored procedures, user-defined functions (UDFs), and triggers.
- Supports views.
- Supports range partitioning, list partitioning, and the definition of multi-level partitions.
- Supports multiple data types. The following table provides a list of data types and their information.

Data type	Alias	Storage size	Range	Description
bigint	int8	8 bytes	-9223372036854775808 to 9223372036854775807	An integer within a large range.
bigserial	serial8	8 bytes	1 to 9223372036854775807	A large auto-increment integer.
bit [(n)]	None	n bits	A bit string constant	A bit string with a fixed length.
bit varying [(n)]	varbit	A bit string with a variable length.	A bit string constant	A bit string with a variable length.

Data type	Alias	Storage size	Range	Description
boolean	bool	1 byte	true/false, t/f, yes/no, y/n, 1/0	A boolean value (true or false).
box	None	32 bytes	((x1,y1),(x2,y2))	A rectangular box on a plane, which is not allowed in a column that is used as the distribution key.
bytea	None	1 byte + binary string	Sequence of octets	A binary string with a variable length.
character [(n)]	char [(n)]	1 byte + n	A string up to n characters in length	A blank-padded string with a fixed length.
character varying [(n)]	varchar [(n)]	1 byte + string size	A string up to n characters in length	A string with a limited variable length.
cidr	None	12 or 24 bytes	None	IPv4 and IPv6 networks.
circle	None	24 bytes	<(x,y),r> (center and radius)	A circle on a plane, which is not allowed in distribution key columns.
date	None	4 bytes	4713 BC - 294,277 AD	Calendar date (year, month, day).
decimal [(p, s)]	numeric [(p, s)]	variable	No limits	User-specified precision, which is exact.
double precision	float8	8 bytes	Precise to 15 decimal digits	Variable precision, which is inexact.
	float			
inet	None	12 or 24 bytes	None	IPv4 and IPv6 hosts and networks.
integer	int, int4	4 bytes	-2.1E+09 to +2147483647	An integer in typical cases.
interval [(p)]	None	12 bytes	-178000000 years - 178000000 years	A time range.
json	None	1 byte + json size	JSON string	A string with an unlimited variable length.
lseg	None	32 bytes	((x1,y1),(x2,y2))	A line segment on a plane, which is not allowed in distribution key columns.
macaddr	None	6 bytes	None	A Media Access Control (MAC) address.

Data type	Alias	Storage size	Range	Description
money	None	8 bytes	-92233720368547758.08 to +92233720368547758.07	Currency amount.
path	None	16+16n bytes	[(x1,y1),...]	A geometric path on a plane, which is not allowed in distribution key columns.
point	None	16 bytes	(x,y)	A geometric point on a plane, which is not allowed in distribution key columns.
polygon	None	40+16n bytes	((x1,y1),...)	A closed geometric path on a plane, which is not allowed in a column that is used as the distribution key.
real	float4	4 bytes	Precise to 6 decimal digits	Variable precision, which is inexact.
serial	serial4	4 bytes	1 to 2147483647	An auto-increment integer.
smallint	int2	2 bytes	-32768 to 32767	An integer within a small range.
text	None	1 byte + string size	A string with a variable length	A string with an unlimited variable length.
time [(p)] [without time zone]	None	8 bytes	00:00:00[.000000] - 24:00:00[.000000]	The time of a day without the time zone.
time [(p)] with time zone	timetz	12 bytes	00:00:00+1359 - 24:00:00-1359	The time of a day with the time zone.
timestamp [(p)] [without time zone]	None	8 bytes	4713 BC - 294,277 AD	The date and time without the time zone.
timestamp [(p)] with time zone	timestamptz	8 bytes	4713 BC - 294,277 AD	The date and time with the time zone.

Data type	Alias	Storage size	Range	Description
xml	None	1 byte + xml size	Variable-length XML string	A string with an unlimited variable length.

11.KVStore for Redis

11.1. What is KVStore for Redis?

KVStore for Redis is a database service that is compatible with open source Redis protocols. KVStore for Redis is based on a highly available hot standby architecture and can scale to meet the requirements of high-performance and low-latency read/write operations.

Features

- KVStore for Redis supports various data types, such as strings, lists, sets, sorted sets, hash tables, and streams. This service also supports advanced features, such as transactions, message subscription, and message publishing.
- KVStore for Redis Enhanced Edition (Tair), which is a key-value pair cloud caching service, is an advanced version of KVStore for Redis Community Edition.

Instance type

Instance type	Overview
Community Edition	KVStore for Redis Community Edition is compatible with the data caching service of Redis-native engines. This service supports master-replica instances and cluster instances.
Performance-enhanced instances of KVStore for Redis Enhanced Edition (Tair)	KVStore for Redis Enhanced Edition (Tair) provides a multi-threading model and integrates some features of Alibaba Tair. KVStore for Redis Enhanced Edition (Tair) supports multiple data structures of Tair and is suitable for diverse scenarios.

11.2. Scenarios

Game industry applications

KVStore for Redis can serve as an important architecture component in the gaming industry.

Scenario 1: Use KVStore for Redis as a storage database

Gaming applications can be deployed in a simple architecture, in which the main program runs on an Elastic Compute Service (ECS) instance and the business data is stored in KVStore for Redis. KVStore for Redis can be used for persistent storage. It uses a master-replica architecture to implement redundancy.

Scenario 2: Use KVStore for Redis as a cache to accelerate connections to applications

You can use KVStore for Redis as a cache to accelerate connections to applications. You can store data in a Relational Database Service (RDS) database that is used as a backend database.

The high availability of KVStore for Redis is essential to your business. If your KVStore for Redis service becomes unavailable, the RDS instances may be overwhelmed by the requests that are sent from your applications. KVStore for Redis adopts the master-replica architecture to ensure high availability. In this architecture, the primary node provides services for your business. If this node fails, the system automatically switches workloads to the secondary node. The complete failover process is transparent.

Live streaming applications

Live streaming service can use KVStore for Redis to store user data and relationship information.

High availability

KVStore for Redis can be deployed in a master-replica architecture to significantly improve service availability.

High performance

KVStore for Redis provides cluster instances to eliminate the performance bottleneck caused by the Redis single-thread mechanism. Cluster instances can effectively handle traffic bursts during live streaming and support high performance.

High scalability

KVStore for Redis allows you to deal with traffic spikes during peak hours by scaling out an instance with a few clicks. The upgrade is completely transparent to users.

E-commerce industry applications

In the e-commerce industry, the KVStore for Redis service is widely used in modules such as commodity presentation and recommendations.

Scenario 1: online shopping systems

An online shopping system is overwhelmed by user traffic during large promotional activities such as flash sales. Most databases cannot handle the heavy load.

To resolve this issue, you can use KVStore for Redis for persistent storage.

Scenario 2: inventory management systems that support stocktaking

KVStore for Redis can be used to count the inventory and RDS can be used to store information about the quantities of items. This way, the KVStore for Redis instance reads count data and the RDS database stores count data. KVStore for Redis is deployed on a physical server. The system provides a high-level data storage capacity based on solid-state drive (SSD) storage that has high performance.

11.3. Benefits

High performance

- Supports cluster instances with a memory capacity of 128 GB or larger. The instances can meet large capacity and high performance requirements.
- Supports master-replica instances with a maximum memory capacity of 32 GB. The instances can meet general capacity and performance requirements.
- Supports CPUs, disks, memory, and network interface controllers (NICs) of different specifications in a cluster without affecting the operational performance of the cluster. This ensures compatibility with your existing devices.

Elastic scaling

- Easy scaling: You can scale the instance storage capacity with only a few clicks by using the console.
- Online scaling: You can scale the instance storage capacity without service interruption.

Resource isolation

- Supports instance-level resource isolation among different instances. This ensures the stability of individual services.
- Supports multi-tenant isolation to ensure that each instance can use exclusive resources, such as CPU, memory, I/O resources, and disks.
- Supports multi-tenant parallel execution on a cluster by using multiple instances. Tasks from tenants are submitted to queues on different instances for execution. KVStore for Redis isolates resources among tenants based on different instances.

High data security

- Data persistence: KVStore for Redis provides high-speed data read/write capabilities and enables data persistence by using a hybrid storage of memory and disks. KVStore for Redis allows you to load data from a persistent database into a cache database.
- Master/replica backup: KVStore for Redis maintains two backup copies of all data on a master node and a replica node to prevent data loss.
- Access control: KVStore for Redis supports password authentication to ensure secure and reliable access to databases.
- Data transmission encryption: KVStore for Redis supports encryption based on SSL and Secure Transport Layer (TLS) to secure data transmission.

High availability

- Master-replica architecture: Each instance runs in a master-replica architecture to eliminate the risk of single points of failure (SPOFs) and ensure high availability.
- Automatic failure detection and recovery: The system automatically detects hardware failures and performs a failover within a few seconds after a failure occurs. This minimizes the adverse impact caused by unexpected hardware failures.
- Supports automatic fault tolerance for server disk failures in a cluster, and supports hot swapping of disks. If a disk fails, services can be recovered within two minutes.

Easy-to-use

- KVStore for Redis is compatible with Redis commands. You can use a Redis client to connect to a KVStore for Redis instance and manage data.
- Supports multiple commands in each query.

Permission management

- Supports data access permission management, such as the logon permissions, table creation permissions, read and write permissions, and whitelist control permissions.
- Allows you to log on to the KVStore for Redis console to manage permissions on access control, including administrative rights settings.
- KVStore for Redis provides a unified permission management feature. This feature allows you to manage various permissions for each component of the system in the KVStore for Redis console. This isolates common users from internal permission management details, simplifies the permission management for administrators, and improves the user experience of permission management.

- Allows you to manage multiple tenants in a centralized manner in the console. For example, you can dynamically configure and manage tenant resources, isolate resources, view statistics on resource usage, and manage tenants at multiple levels.

Scheduling

Supports multi-cluster scheduling, multi-resource pool scheduling, and multi-tenant scheduling.

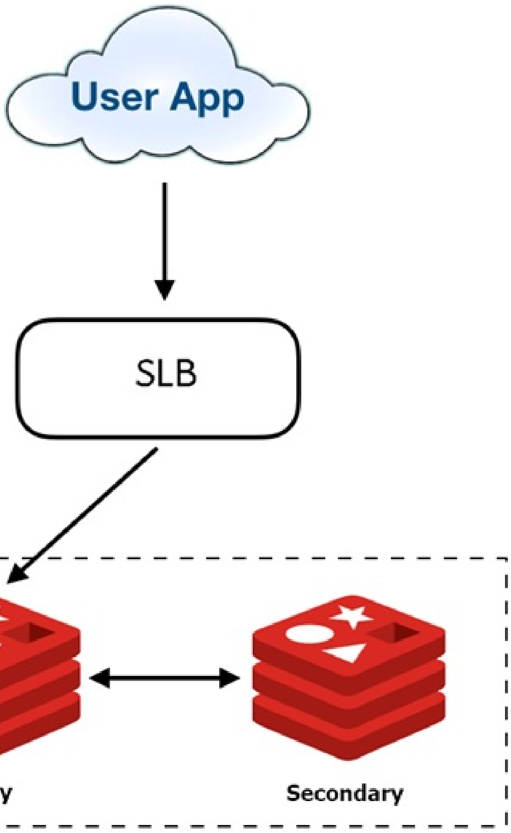
11.4. Architectures

11.4.1. System architectures

KVStore for Redis provides two types of architecture: the master-replica architecture and the cluster architecture.

Master-replica instances

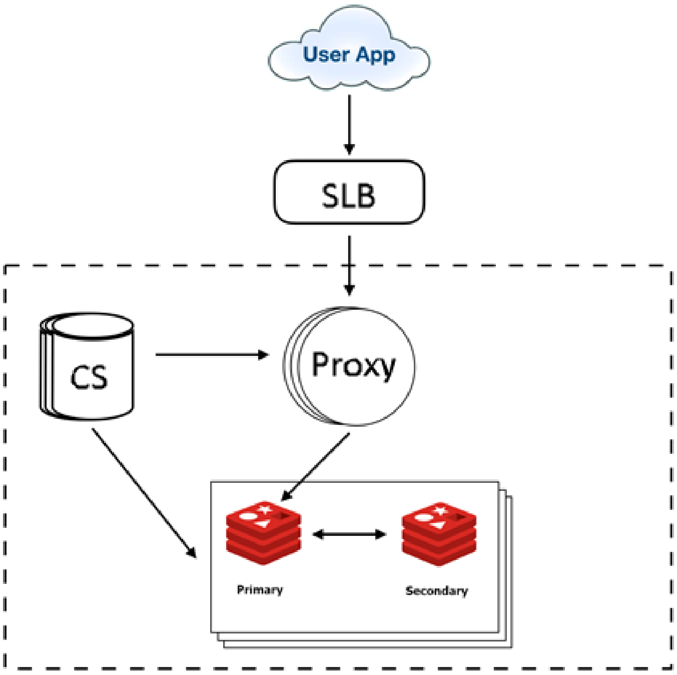
The following figure shows the master-replica architecture.



The master-replica architecture consists of a master node and a replica node. You can directly access the master node through an Server Load Balancer (SLB) connection.

Cluster instances

The following figure shows the cluster architecture.



The cluster architecture consists of three components: redis-config (cs), redis-proxy (proxy), and the Redis kernel.

The cluster architecture consists of multiple cs nodes, proxy nodes, and master/replica Redis nodes. After you access a proxy node through an SLB connection, the proxy node forwards your request to the corresponding shard of a master Redis node.

11.4.2. Components

This topic describes the components of KVStore for Redis and how these components work.

Redis-config

The redis-config (cs) component stores the metadata and topology information of clusters. Operations and maintenance (O&M) tasks are performed by using the cs component. The cs component maintains a heartbeat connection with the proxy component and the Redis kernel, and synchronizes metadata and topology information of clusters to the proxy component and the Redis kernel.

Redis-proxy

The redis-proxy (proxy) component is the proxy server that connects a client to a Redis server and implements Redis protocols. The proxy component can verify user permissions, forward requests, provide collect monitoring data at an interval of several seconds.

Redis kernel

The Redis kernel is optimized based on the open-source Redis kernel, and provides higher performance and more features.

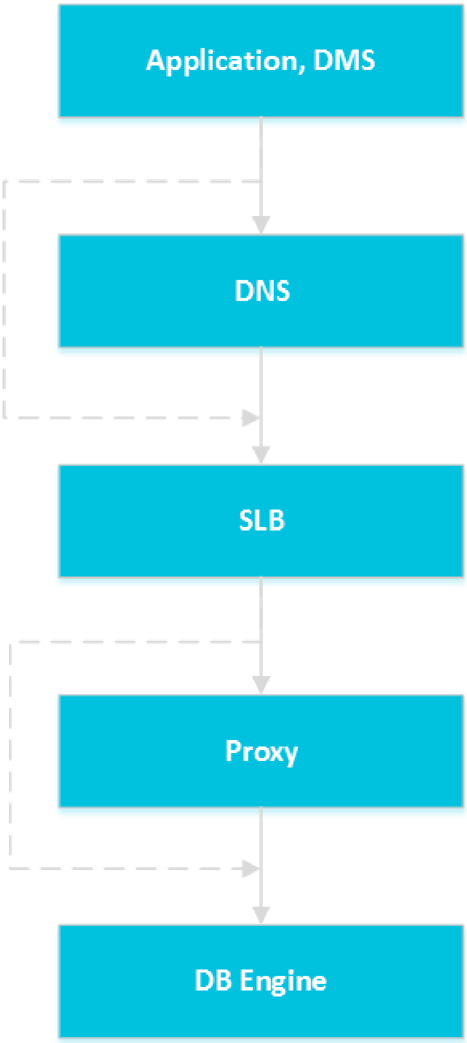
11.5. Features

11.5.1. Data link service

11.5.1.1. Overview

The data link service allows you to add, delete, modify, and search data.

You can connect to the KVStore for Redis service by using your application.



11.5.1.2. DNS

The Domain Name System (DNS) module can dynamically resolve domain names to IP addresses. Therefore, IP address changes cannot affect the performance of KVStore for Redis.

For example, the domain name of an KVStore for Redis instance is test.kvstore.aliyun.com, and the IP address corresponding to this domain name is 10.1.1.1. You can connect to the KVStore for Redis instance if you add test.kvstore.aliyun.com or 10.1.1.1 to the connection pool of your application.

If you migrate the KVStore for Redis instance to another host after a failure occurs or upgrades the instance version, the IP address may change to 10.1.1.2. You can connect to the KVStore for Redis instance if you add test.kvstore.aliyun.com to the connection pool of your application. However, if you add 10.1.1.1 to the connection pool, you cannot connect to the instance.

11.5.1.3. SLB

The Server Load Balancer (SLB) module can forward traffic to available instance IP addresses. Therefore, physical server changes cannot affect the performance of KVStore for Redis.

For example, the private IP address of an KVStore for Redis instance is 10.1.1.1, and the corresponding proxy module or database engine module runs on 192.168.0.1. Typically, the SLB module forwards all traffic destined for 10.1.1.1 to 192.168.0.1. When the proxy module or database engine module fails, the hot standby proxy module or database engine module with the IP address 192.168.0.2 takes over. The SLB module redirects access traffic from 10.1.1.1 to 192.168.0.2 and the KVStore for Redis instance continues to run normally.

11.5.1.4. Proxy

The Proxy module provides some features such as data routing, traffic detection, and session persistence.

- Data routing: supports partition policies and complex queries for distributed routes based on a cluster architecture.
- Traffic detection: reduces the risks from cyberattacks that exploit Redis vulnerabilities.
- Session persistence: prevents connection interruptions in the case of failures.

11.5.1.5. DB Engine

KVStore for Redis supports standard protocols.

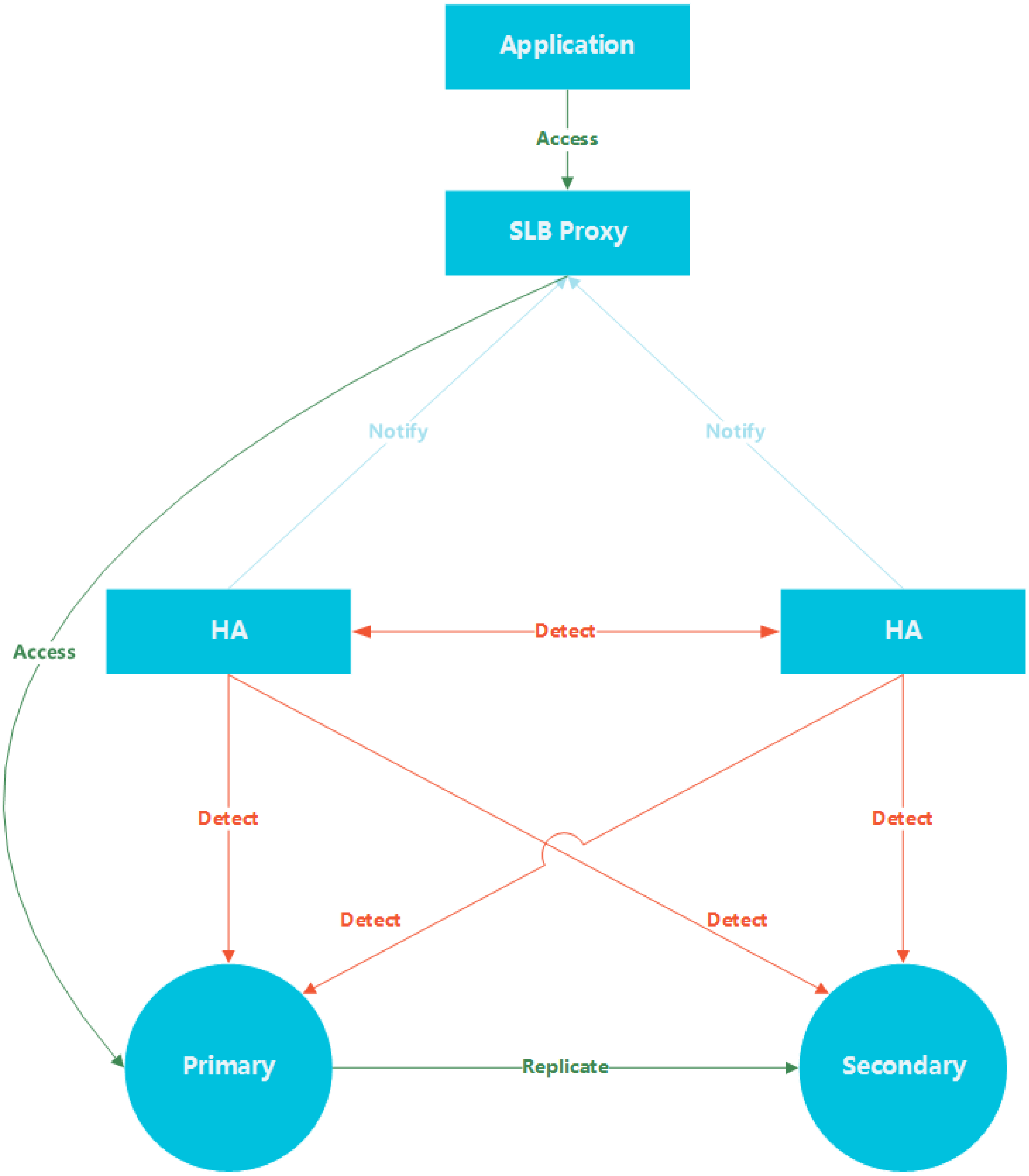
Engine	Version
Redis	2.8, 4.0, and 5.0

11.5.2. HA service

11.5.2.1. Overview

The high-availability (HA) service guarantees the availability of data link services and handles internal database exceptions.

The HA service is also highly available because this service contains multiple HA nodes.



11.5.2.2. Detection

The Detection module checks whether the primary and secondary nodes of the database engine are operating normally.

An HA node receives the heartbeat from the primary database engine node at an interval of 8 to 10 seconds. This information, combined with the heartbeat information of the secondary and other HA nodes, allows the Detection module to eliminate false negatives and positives caused by exceptions such as network jitter. As a result, switchover can be completed within 30 seconds.

11.5.2.3. Repair

The Repair module maintains replications between the primary node and the secondary node of DB Engine. This module also fixes errors that occur on either node during normal operations as follows:

- Automatically fixes exceptionally disconnected replications between these nodes.
- Automatically fixes table-level damages on both nodes.
- Automatically saves crash events and fixes the failures on both nodes.

11.5.2.4. Notice

The Notice module notifies the SLB or Proxy module of status changes of primary and secondary nodes. Therefore, you can connect to available nodes.

For example, the Detection module locates an exception on a primary node and notifies the Repair module to fix the exception. If the Repair module fails to resolve the issue, the Repair module notifies the Notice module to perform failover. Afterward, the Notice module forwards the failover request the Server Load Balancer (SLB) or Proxy module to switch all traffic to the secondary node. Meanwhile, the Repair module creates a secondary node on a different physical server and synchronizes this change to the Detection module. The Detection module checks the health status of the instance again to verify that the instance is healthy.

11.5.3. Monitoring service

11.5.3.1. Service-level monitoring

The service module tracks the status of services. The service module of KVStore for Redis monitors the status of cloud services that KVStore for Redis depends on, such as Server Load Balancer (SLB). The monitored metrics include features and response time.

11.5.3.2. Network-level monitoring

The network module tracks the network status. The monitored metrics include:

- Connectivity between Elastic Compute Service (ECS) and KVStore for Redis
- Connectivity between physical servers of KVStore for Redis
- Packet loss rates on vRouters and vSwitches

11.5.3.3. OS-level monitoring

The operating system (OS) module traces status of hardware and the kernel of an operating system. The monitoring metrics include:

- Hardware inspection: the OS module regularly checks the running status of devices such as CPUs, memory modules, motherboards, and storage devices. When locating any potential hardware failures, the module automatically raises a request for repair.
- OS kernel monitoring: the OS module traces all kernel requests for databases, and analyzes the cause of a slow or error response to a request according to the kernel status.

11.5.3.4. Instance-level monitoring

The instance module of KVStore for Redis collects instance-level information. The monitored metrics include:

- Instance availability
- Instance capacity

11.5.4. Scheduling service

The scheduling service activates and migrates instances for users by allocating and integrating underlying resources for KVStore for Redis.

For example, if you use the console to create an instance, the scheduling service will select the optimal physical server to handle the instance traffic.

After a large number of instance creation, deletion, and migration operations are performed over an extended period of time, resource fragments are generated in the data center. The scheduling service calculates the degree of resource fragmentation and periodically integrates resources to increase the service capacity of the data center.

12.ApsaraDB for MongoDB

12.1. What is ApsaraDB for MongoDB?

ApsaraDB for MongoDB is a stable, reliable, and resizable database service fully compatible with MongoDB protocols. ApsaraDB for MongoDB offers a full range of database solutions, such as disaster recovery, backup, restoration, monitoring, and alerts.

ApsaraDB for MongoDB supports the following features:

- Incorporates advanced functions such as disaster recovery failover and downtime migration.
- Supports quick database backup and restoration. You can easily perform standard database backup and database rollback operations in the ApsaraDB for MongoDB console.
- Provides over 20 performance monitoring metrics and alerting. This helps you learn about the performance status of your database.
- Provides visual data management tools for convenient O&M.
- Enables you to create instances on VPCs.

12.2. Benefits

High availability

- The three-node replica set and sharded cluster architectures that deliver extremely high service availability

  ApsaraDB for MongoDB uses high-availability architectures of three-node replica sets and sharded clusters. The data nodes are located on different physical servers and can automatically synchronize data.

- Automatic backup and quick data restoration

  Data is automatically backed up and uploaded to Object Storage Service (OSS) on a daily basis. This improves data disaster recovery capabilities and reduces disk space consumption. Backup files can be used to restore instance data to their original instance. This prevents irreversible effects on service data caused by incorrect operations and errors.

Data backup

ApsaraDB for MongoDB allows you to perform full or incremental backup and restore data from storage. Clusters can be backed up between data centers. The backup process is visually presented.

High security

- Anti-DDoS protection: ApsaraDB for MongoDB filters inbound traffic. When DDoS attacks are identified, the source IP addresses will be scrubbed. If scrubbing fails, blackhole filtering is triggered.
- IP whitelist configuration: You can configure up to 1,000 IP addresses to connect to an ApsaraDB for MongoDB instance. The IP whitelist can directly control risks at the source.

Audit log management

You can store audit logs in Log Service and download logs from Log Service. This facilitates long-term storage and management of audit logs.

Ease of use

ApsaraDB for MongoDB provides various performance monitoring features. ApsaraDB for MongoDB provides real-time monitoring information about the CPU utilization, connections, and disk usage of your instances and sends alerts to keep you informed of the status of your instances.

Scalability

Both replica set and sharded cluster instances of ApsaraDB for MongoDB support elastic scaling. You can change the configuration of your instance if the current configuration cannot meet performance requirements or is unsuitable for your business needs. The configuration change process is completely transparent and does not affect your business.

Isolation

ApsaraDB for MongoDB uses multiple instances and implements tasks for multiple tenants within a cluster at the same time. The tasks of the tenants are implemented in different instances. ApsaraDB for MongoDB instances are separated to isolate resources between different tenants.

Permission management

ApsaraDB for MongoDB allows you to configure and manage tenants in a dynamic and centralized manner. You can also isolate resources and query usage statistics of resources. Management of multi-level tenants is supported.

Scheduling

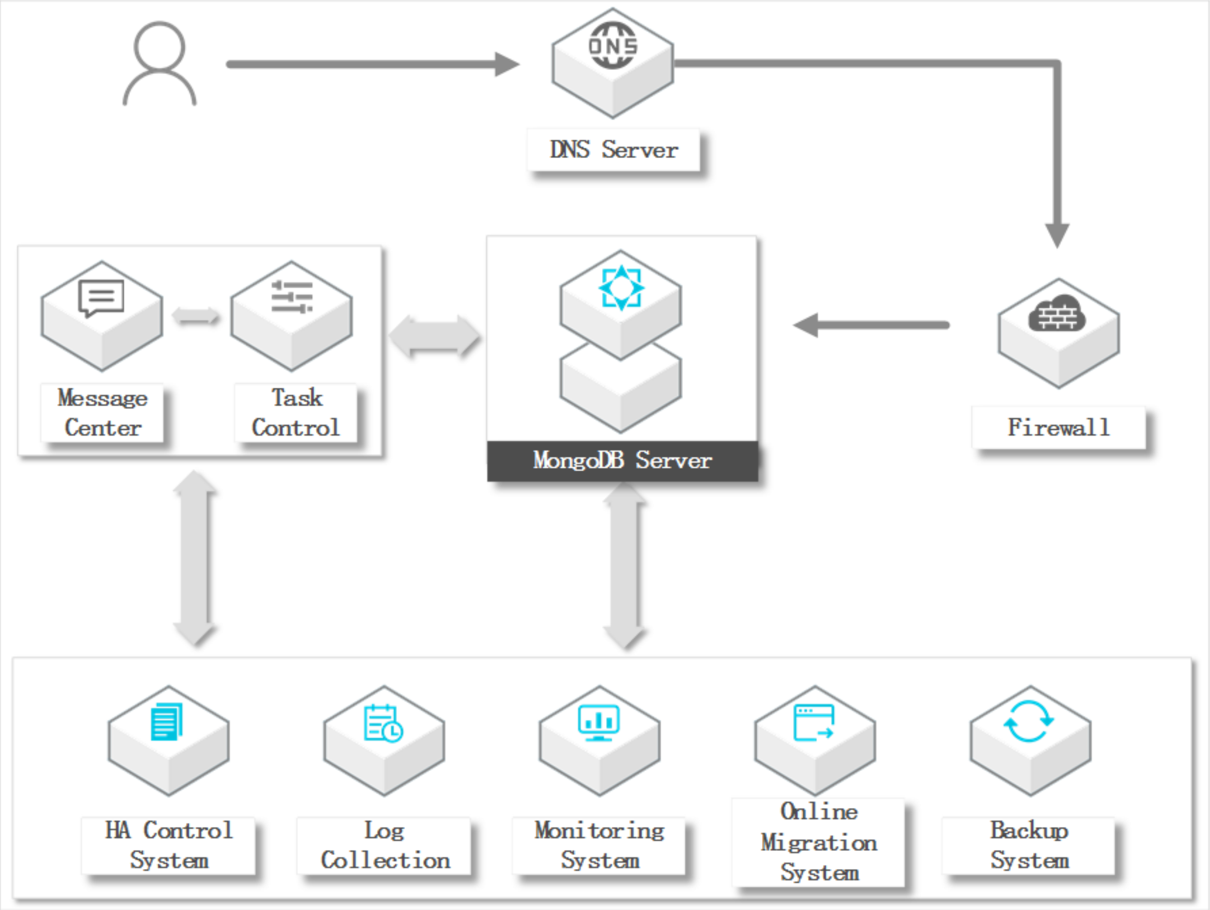
ApsaraDB for MongoDB schedules resources from multiple resource pools for tenants within different clusters.

12.3. System architecture

12.3.1. ApsaraDB for MongoDB system architecture

This topic describes the architectures and components of ApsaraDB for MongoDB.

The following figure shows the ApsaraDB for MongoDB system architecture.

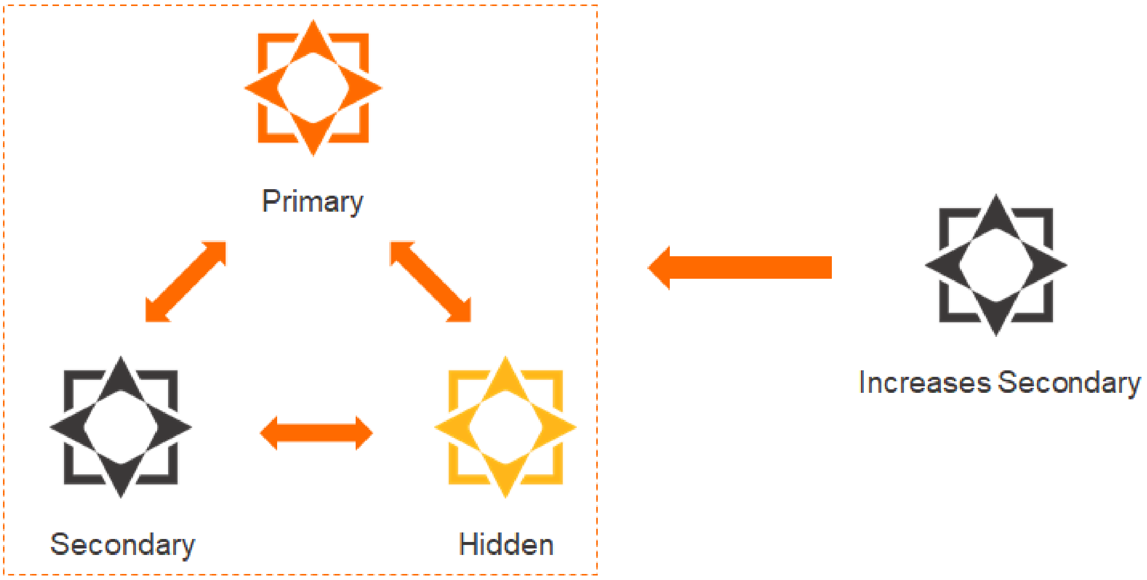


- **HA control system**: This module checks the high availability of an instance. You can use this module to detect and monitor the running status of ApsaraDB for MongoDB instances. If the system determines that the primary node of an ApsaraDB for MongoDB instance is unavailable, it fails over to a secondary node to ensure the high availability of the instance.
- **Log collection**: This module collects MongoDB running logs, including the slow query log and access log of an instance.
- **Monitoring system**: This module collects performance monitoring information about MongoDB instances, including basic metrics, disk capacity, network requests, and the number of operations that you have performed.
- **Online migration system**: When an error occurs with the physical server that hosts the instances, this module recreates an instance on the fly based on the backup files stored in the backup system. This ensures the continuity of your business.
- **Backup system**: This module generates ApsaraDB for MongoDB instance backups and stores the backup files in OSS. This module allows you to configure custom backup settings and create temporary backups. Files are retained for seven days.
- **Task control**: ApsaraDB for MongoDB supports multiple management operations, such as creating instances, changing instance configurations, and backing up instances. This module controls and tracks these tasks and troubleshoots errors based on your commands.

12.3.2. Architecture of replica set instances

Replica set instances use the three-node architecture. This topic describes the three nodes of a replica set instance.

Architecture



ApsaraDB for MongoDB uses a multi-node architecture to ensure high availability. A replica set instance consists of a primary node, one or more secondary nodes, and a hidden node. You can manage primary and secondary nodes. The following section describes the three nodes of a replica set instance:

- Primary node: All read and write operations are performed on the primary node. Each replica set instance contains only one primary node.
- Secondary node: The data of a secondary node is synchronized with the primary node by using oplogs. If the primary node fails, a secondary node can be elected as the new primary node to ensure high availability.

  > **Note** When you connect to an instance by using the endpoint of a secondary node, you can only read data from the instance but cannot write data to it.

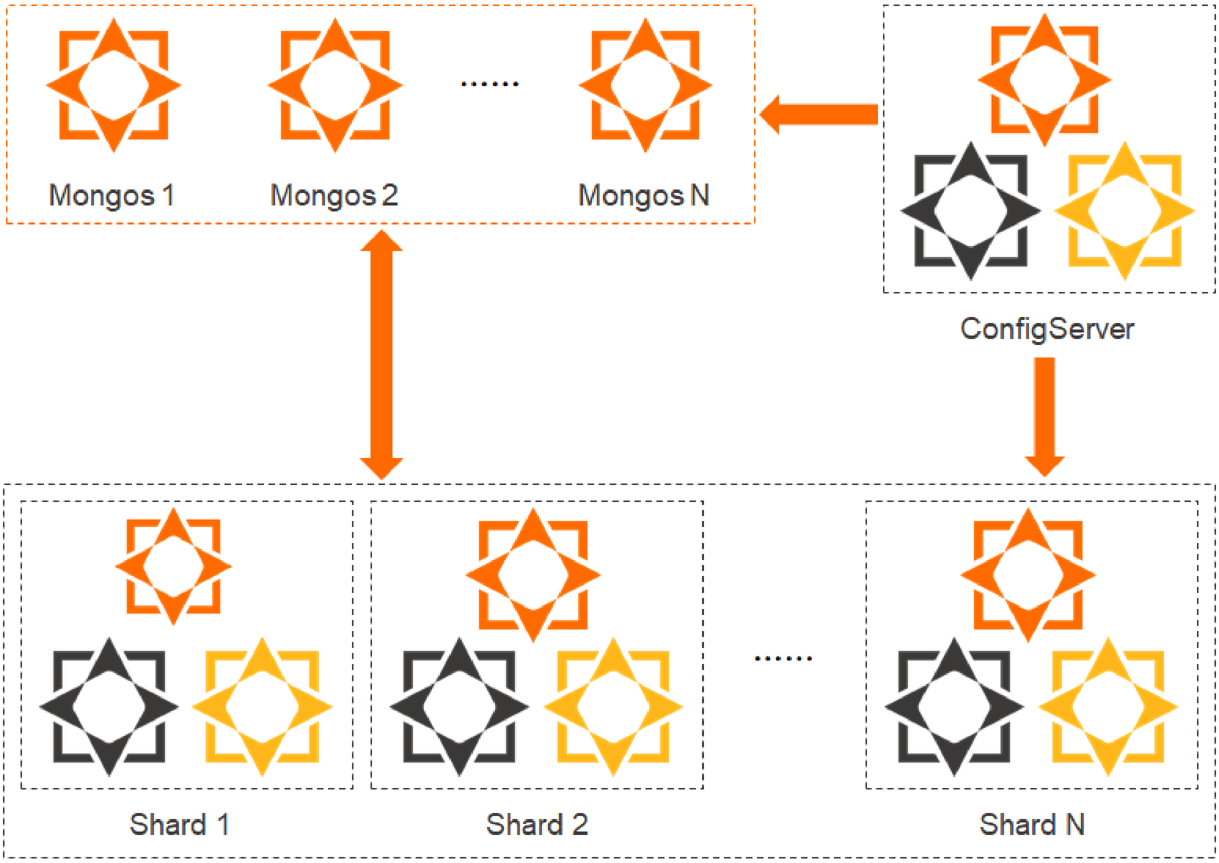
- Hidden node: The data of a hidden node is synchronized with the primary node by using oplogs. If a secondary node fails, the hidden node can be elected as the new secondary node to ensure high availability.

  > **Note** The hidden node is used only to ensure high availability and is not visible to users.

12.3.3. Architecture of sharded cluster instances

A sharded cluster instance consists of three components: mongos, shard, and Configserver. This topic describes the components of a sharded cluster instance.

Architecture



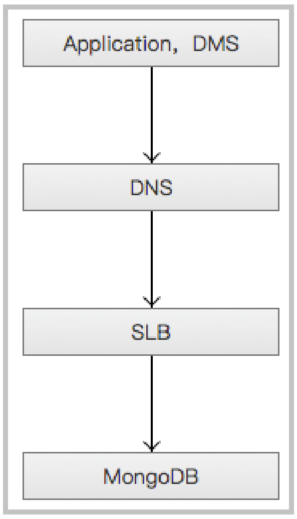
Components

Component	Architecture	Description
Mongos	Standalone architecture	Queries and writes routes to the corresponding shard node.
Shard	Replica set architecture (three nodes)	Stores database data.
Configserver	Replica set architecture (three nodes)	Stores the metadata for clusters and shards. The metadata contains information about which data is stored in each shard. **Note** You cannot change the specifications of the Configserver (1 vCPU, 2 GiB memory, and 20 GB storage capacity).

12.4. Features

12.4.1. Data link service

This topic describes the data link service, which allows you to perform operations on data.



DNS

For example, the endpoint of a node in an ApsaraDB for MongoDB instance is mongodb.aliyun.com, and the IP address that corresponds to this endpoint is 10.1.1.1. To connect an application to the instance, you can create a connection to mongodb.aliyun.com or 10.1.1.1 in the connection pool.

However, the IP address may change to 10.1.1.2 if the instance is upgraded or migrated. In this case, if mongodb.aliyun.com is configured in the connection pool, the application can still access the instance. If 10.1.1.1 is configured in the connection pool, the application can no longer access the instance.

SLB

The SLB module uses both the private and public IP addresses of an ApsaraDB for MongoDB instance, so server changes do not affect the performance of the instance.

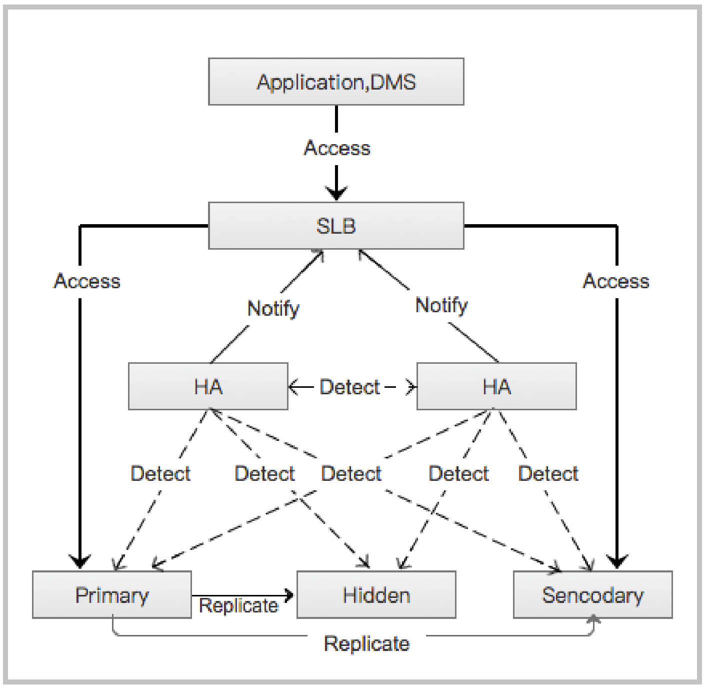
For example, the private IP address of a node in an ApsaraDB for MongoDB instance is 10.1.1.1, and this ApsaraDB for MongoDB instance actually runs on a server whose IP address is 192.168.0.1. Typically, the SLB module forwards all traffic destined for 10.1.1.1 to 192.168.0.1.

If the server with the IP address 192.168.0.1 fails, another server in the hot standby state with the IP address 192.168.0.2 takes over services from the server with the IP address 192.168.0.1. The SLB module then redirects all traffic destined for 10.1.1.1 to 192.168.0.2.

12.4.2. High availability service

The high availability (HA) service guarantees the availability of data link services and handles internal database exceptions.

In addition, this service is based on multiple HA nodes that are also highly available.



Detection

The Detection module detects the running or faulty status of the primary, secondary, and hidden nodes for ApsaraDB for MongoDB. An HA node uses heartbeat information, which is acquired at an interval of 8 to 10 seconds, to determine the health status of the primary node. This information, combined with the heartbeat information of the secondary and hidden nodes, allows the Detection module to eliminate any risk of false negatives and positives caused by exceptions such as network jitters. Switchover can be completed quickly.

Repair

The Repair module maintains the replication relationship among the primary, secondary, and hidden nodes, and fixes faulty nodes or creates new nodes.

Notice

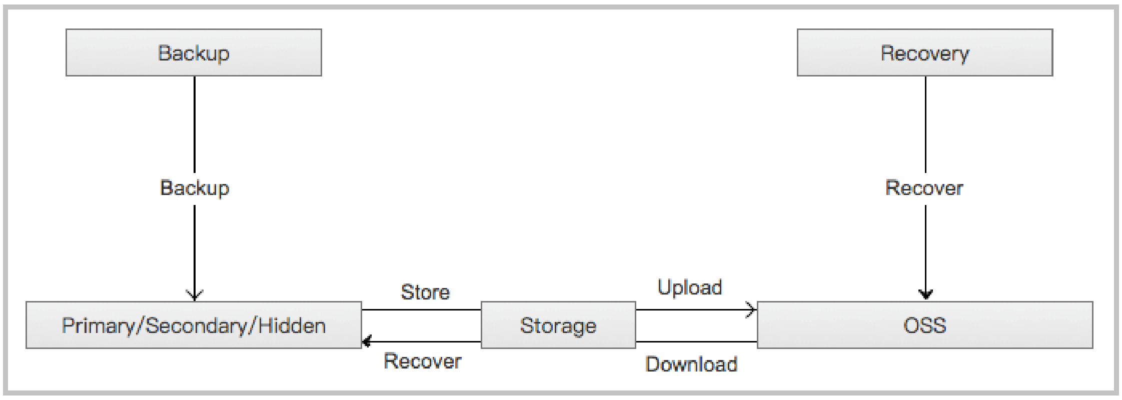
The Notice module informs SLB of node status changes to ensure that you can access the available node.

For example, the Detection module will instruct the Notice module to switch traffic if the Detection module discovers that an exception occurs with the primary node. The Notice module then forwards the switched traffic request to SLB, which redirects traffic from the primary node to the secondary node or from the secondary node to the hidden node. In this circumstance, the secondary node becomes the primary node and the hidden node becomes the secondary node.

During this process, the Repair module attempts to fix the original primary node and convert it to a new hidden node. If the Repair module fails to fix the original primary node, the Repair module will create a new hidden node on another physical server and synchronize the change to the Detection module. The Detection module then incorporates the information and rechecks the health status of the instance.

12.4.3. Backup service

The backup service supports offline data backup, transfer, and recovery.



Backup

The Backup module backs up and compresses data and logs of an instance, and uploads the compressed files to OSS. Data backup in ApsaraDB for MongoDB is performed on the hidden node to avoid affecting services on the primary and secondary nodes.

Recovery

The Recovery module restores backup files stored in OSS to a specified node.

Primary node rollback: You can roll back the settings on the primary node to a specific point in time if you mistakenly perform operations on data.

Secondary and hidden node restore: The system automatically selects a new secondary node to reduce risks when an irreparable failure occurs with the original secondary node.

Storage

The Storage module uploads, dumps, and downloads backup files. Currently, all backup data is uploaded to OSS for storage. You can obtain temporary links to download the data as needed.

12.4.4. Monitoring service

The monitoring service tracks the status of services, networks, operating systems, and instances.

Service

The Service module tracks the status of Alibaba Cloud services. For example, the Service module can monitor SLB, OSS, and SLS services and check whether their functions work as expected and the response time is acceptable. ApsaraDB for MongoDB is dependent on these services. The module also uses corresponding logs to check whether the internal services of ApsaraDB for MongoDB are running properly.

Network

The Network module tracks the status of networks. For example, the Network module can monitor the connectivity between ECS and ApsaraDB for MongoDB instances and among ApsaraDB for MongoDB physical machines. It can also monitor packet loss rates of VRouters and VSwitches.

OS

The OS module tracks the status of hardware and OS kernels.

Examples:

- Hardware inspection: The OS module regularly checks the running status of components such as CPUs, memory modules, motherboards, and storage devices. If the module detects any potential hardware failures, it automatically submits a repair ticket.
- OS kernel monitoring: The OS module tracks all kernel invocations of databases and analyzes the cause of a slow or faulty invocation based on the kernel status.

Instance

The Instance module supports the following features:

- Collects ApsaraDB for MongoDB instance information.
- Provides instance availability information.
- Monitors instance capacity and performance metrics.
- Records statement executions for instances.

12.4.5. Scheduling service

The scheduling service allocates resources and manages instance versions.

Resource

The Resource module allocates and integrates underlying ApsaraDB for MongoDB resources. This module allows you to create and modify instances.

For example, when you use the ApsaraDB for MongoDB console or API to create an instance, the Resource module will calculate and then select the most suitable server to handle the network traffic. After you have created, deleted, and migrated instances multiple times, the Resource module can calculate the resource fragmentation rate in a zone and periodically integrates the resources to improve service capabilities of the zone.

Version

The Version module allows you to upgrade ApsaraDB for MongoDB instances. For example, you can upgrade an ApsaraDB for MongoDB instance to a major version, such as from version 3.2 to version 3.4. You can also upgrade an instance to a minor version that has optimized the source code or kernel as required.

12.4.6. Migration service

The migration service enables you to migrate data from a user-created database to ApsaraDB for MongoDB.

Data Transmission Service (DTS) is a data stream service provided by Alibaba Cloud for data exchanges between data sources. It supports full and incremental migration for ApsaraDB for MongoDB.

- Full data migration: DTS migrates all data from source databases to destination instances.
- Incremental data migration: During incremental migration, the updated data in the local MongoDB database is synchronized to an ApsaraDB for MongoDB instance. Ultimately, the local MongoDB database and the ApsaraDB for MongoDB instance enter the dynamic synchronization process. Incremental migration enables data migration from a local MongoDB database to an ApsaraDB for MongoDB instance without interrupting the services provided by the local MongoDB database.

13.Data Management (DMS)

13.1. What is DMS?

Data Management (DMS) is a database management service that is provided by Alibaba Cloud. This service allows you to manage relational databases, such as MySQL, SQL Server, PolarDB-X (previousely called DRDS), PostgreSQL, Oracle, and ApsaraDB for OceanBase. This service also allows you to manage NoSQL databases such as MongoDB. DMS is a fully managed data management service. You can use DMS to view BI charts and data trends, track data, optimize performance, and implement access control. You can also use DMS to manage data, schemas, and servers.

- DMS provides support for the entire process of database development. The process includes the following stages: 1. Design table schemas 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.
- DMS supports field-level access control. All operations on databases are synchronized to online environments and can be traced.
- DMS allows you to configure different approval processes for different operations. These operations include schema design, data changes, and data export.
- DMS integrates database development and interaction. When you manage databases in DMS, you do not need an account and a password to connect to a database or switch between multiple databases.
- DMS detects operational changes and identify risks. Database administrators (DBAs) can control risks by classifying operations into multiple levels based on previous experience.

13.2. Product value

DMS provides an easy-to-use and secure database access and management platform. Visualized data services allow you to use databases on browsers. This eliminates the need to install various database clients. When you edit data, you can perform operations on table data and change table schemas with ease, without the need to write complex SQL statements. DMS provides advanced features that are not provided by common clients. These advanced features allow you to synchronize table schemas, create development processes, and create development specifications.

Before you use DMS, you must first log on to the Apsara Uni-manager Management Console. Then, you can use your database account and password to log on to DMS by using two-factor authentication. This prevents your database account and password from being stolen. DMS supports data transmission over HTTS or SSL. This feature prevents data from being intercepted or tampered during data transmission.

DMS also supports RAM and STS for permission verification. This prevents against unauthorized access.

DMS supports access to VPC instances. It provides you with an interface for data access and ensures the network security of database instances. Common clients do not provide this feature.

DMS is an industry-leading database DevOps solution for enterprises. This solution implements secure access control on enterprise core data. This also ensures secure and efficient management of databases. DMS allows you to improve the efficiency of collaboration between developers and DBAs.

13.3. Benefits

DMS provides multiple benefits. These benefits include various data sources, secure and controllable processes, and fine-grained permission management. These benefits allow you to improve data security and simplify data management.

Various data sources

- Relational databases: MySQL, SQL Server, PostgreSQL, PolarDB-X (previously called DRDS), Oracle, and ApsaraDB for OceanBase.
- NoSQL databases: ApsaraDB for Redis and ApsaraDB for MongoDB.
- Analytical databases: AnalyticDB for MySQL and AnalyticDB for PostgreSQL.

Unified operations and comprehensive audits

- After you adds a database instance to DMS as an administrator, you can perform the required operations in the DMS console. These operations include querying databases, changing schemas, and changing data.
- You can query and audit all historical operations based on multiple dimensions. These dimensions include the operator, database, table, and time.

Fine-grained access control

Common users do not need to use database accounts and passwords. These users only need to request the query, export, or change permission on the destination database, table, or field in the DMS console based on their business requirements. After a permission expires, DMS revokes the permission.

Custom approval processes

You can create custom approval processes for the modules of each database instance. These approval processes are specific to your business requirements. This allows you to meet requirements from several aspects, such as efficiency and security. Example:

- Impose loose controls on a test environment. You can reduce stages or set no approval process.
- Impose strict controls on a production environment. You can specify an approval process that includes the required operations for the production environment. The production environment takes effect until all these operations are approved by the specified engineers in sequence.

Custom design specifications for schemas

You can create custom design specifications for MySQL table schemas. These design specifications include the field type, index type, number of indexes, field length, table size, and release process.

Simple procedure to schedule and orchestrate periodical tasks

DMS provides a quick method to create the required orchestration and recurring schedules for the SQL task nodes of various databases in a quick manner. You can use this feature to perform serval operations on databases to explore the value of data. These operations include transferring historical data and generating periodical reports.

13.4. Architecture

DMS provides database management services in the business layer, scheduling layer, and connection layer. This architecture allows DMS to handle real-time access to databases and schedule backend data tasks.

13.5. Features

13.5.1. Workbench

Category	Description
Feature	This feature provides a user center. This user center provides multiple management shortcuts that direct you to the related pages. These shortcuts allow you to view the tickets, recent databases, followed databases, authorized databases, and owned databases.
Scenario	Query tickets and databases, manage user permissions, and control access to owned databases.
Procedure	Log on to the DMS console, go to the **Workbench** tab, and then use the required feature.
Limit	No limit is set.

13.5.2. Top search box

Category	Description
Feature	This feature allows you to query all the databases or tables of DMS.
Scenario	Query databases or tables to perform the required operations.
Procedure	Query databases or tables by name, keyword, or a combination of conditions
Limit	The control mode of the required instance must be security collaboration.

13.5.3. Permission application

Category	Description
Feature	This feature allows you to apply for the required permissions. • For example, you can request the query, export, or change permission for a database, table, field, or row. • You can request the logon permission for an instance. • You can request the ownership of an instance, database, or table.
Scenario	Grant permissions to common users.

Category	Description
Procedure	In the top navigation bar of the DMS console, choose **Task Type > Permission Application** and submit the required application.
Limit	To apply a permission to the database, table, field, or row of an instance, the control mode of the instance must be security collaboration.

13.5.4. Data Plans

13.5.4.1. Data Changes

Category	Description
Feature	This feature allows you perform multiple operations to change the destination database or table, for example, data definition language (DDL) and data manipulation language (DML) operations. These operations include changing common data, clearing historical data, changing data without locking tables, changing programmable objects, and exporting data in batches.
Scenario	Initialize schemas, initialize data, and rectify data during the development of a project or after the code of the project is released.
Procedure	In the top navigation bar of the DMS console, choose **Data Plans > Data Changes** and submit the required SQL script to be executed.
Limits	The operator must have the change permission on the destination database or table.

13.5.4.2. Change schemas without locking tables

Category	Description
Feature	This feature allows you to update the schemas of a table without the need to lock the table. This prevents business interruption or the latency issue of the secondary database due to locked tables.
Scenario	Update table schemas.
Procedure	In the top navigation bar of the DMS console, choose **System > Instance**, find the required instance, and then click Edit next to the instance. In the Edit instance dialog box, click **Advanced information** to enable the change schemas without the need to lock tables feature. Use one of the procedures to change the schemas of a table without the need to lock the table: • In the top navigation bar of the DMS console, choose **Schemas > Schema Design**. • In the top navigation bar of the DMS console, choose **Data Plans > Data Changes > Normal Data Modify**. • In the top navigation bar of the DMS console, choose **System Management > Task** and submit the required SQL change task.

Category	Description
Limit	- The operator must have the change permission on the destination database. - The control mode of the required instance must be security collaboration.

13.5.4.3. Data import

Category	Description
Feature	This feature provides a quick method for you to import a large amount of data to databases. This reduces the costs of labors and material resources.
Scenario	Import a large amount of data.
Procedure	In the top navigation bar of the DMS console, choose **Data Plans > Data Import**. Export the required contents, and submit an executable SQL script or a CSV file.
Limit	- The operator must have the change permission on the destination database. - The destination database must be one of the following databases: MySQL, ApsaraDB for OceanBase, and PolarDB-X.

13.5.4.4. Data export

Category	Description
Feature	This feature allows you to export data from the destination database or table. The data include SQL result sets, databases, and tables.
Scenario	Analyze a large amount of data.
Procedure	In the top navigation bar of the DMS console, choose **Data Plans > Data Export**, and export the required contents or submit an executable SQL script.
Limit	The operator must have the export permission on the destination database or table.

13.5.4.5. Test data generation

Category	Description
Feature	This feature allows you to create a custom data generation method based on the data type of a field. We recommend that you use this feature. This feature can also be used to generate a large amount of test data. This reduces the time that is required to generate test data.
Scenario	Prepare data for a test environment.

Category	Description
Procedure	In the top navigation bar of the DMS console, choose **Data Plans > Test Data Generate** and configure the required data to be generated.
Limits	• The operator must have the change permission on the destination database or table. • The destination database must be a relational database, such as MySQL.

13.5.4.6. Database cloning

Category	Description
Feature	This feature allows you to synchronize the schemas and data of a source database to a destination database.
Scenario	Create schemas and data for a test environment.
Procedure	In the top navigation bar of the DMS console, choose **Data Plans > Database Clone** and submit a ticket to clone the destination database.
Limit	• The operator must have the change permission on the destination database. • The destination database must be a relational database, such as MySQL.

13.5.5. Data factory

13.5.5.1. Task orchestration

Category	Description
Feature	This feature allows you to orchestrate SQL and DSQL nodes of various database types based on a directed acyclic graph (DAG) workflow. You can add these nodes to periodic scheduling tasks and perform the required operations. For example, you can develop data warehouses, transform periodical report, or transfer historical data. You can also use this feature to implement data integration, deploy SQL task nodes, check dependencies, and implement the periodic scheduling of sample task flows.
Scenario	Analyze periodical data and generate reports.
Procedure	In the top navigation bar of the DMS console, choose **Data Factory > Task Orchestration**, and configure the required task nodes to orchestrate. Then, configure periodic scheduling tasks.
Limit	• The operator must have the change permission to the destination database or table. • The control mode of the related instance must be security collaboration.

13.5.5.2. Data warehouse development

Category	Description
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.
Scenario	Implement data integration, data transformation, data visualization, and data mining.
Procedure	In the top navigation bar of the DMS console, choose **Data Factory > Data Warehouse Development** and configure the required information. The information includes the data warehouse project and task process.
Limit	• The operator must assume the owner role of the destination database. • The control mode of the related instance must be security collaboration.

13.5.5.3. Data services

Category	Description
Feature	This feature allows you to define API operations based on the data that is managed in DMS. You can call the required API operations to export the managed data in a quick manner.
Scenario	Export data of minimum granularity, generate visualized data, and provide transformed data that can be read by multiple applications.
Procedure	In the top navigation bar of the DMS console, choose **Data Factory > Data Service** and perform the required operations on an API operation. For example, you can develop, deactivate, test, or release an API operation.
Limit	• The operator must have the query permission to the destination database or table. • The control mode of the related instance must be security collaboration.

13.5.6. Schemas

13.5.6.1. Synchronization between tables and databases

Category	Description
Feature	• This feature allows you to compare the differences between the source table and the destination table. Then, you can submit and execute differential data definition language (DDL) scripts. • This feature provides multiple tools. You can use these tools to initialize empty databases, synchronize schemas, and repair table consistency.
Scenario	Synchronize table schemas.

Category	Description
Procedure	In the top navigation bar of the DMS console, choose **Schemas > Table Sync**, and submit the required comparison task. Then, execute a diff script.
Limit	• The operator must have the query permission to the source database and the change permission to the destination database. • The source and destination databases must be MySQL or ApsaraDB for OceanBase databases.

13.5.6.2. Schema design

Category	Description
Feature	This feature allows you to design table schemas for the destination databases or tables. These table schemas conform to the predefined development specifications
Scenario	Design table schemas during the development of a project.
Procedure	In the top navigation bar of the DMS console, choose **Schemas > Schema Design**and edit the required table schema. Then, generate a change script and apply the changes to the destination database by executing the change script.
Limit	• The operator must have the change permission to the destination database. • The destination database must be a MySQL or ApsaraDB for OceanBase database. • The control mode of the related instance must be security collaboration.

13.5.7. SQL reviews

Category	Description
Feature	This feature allows you to review SQL statements in code.
Scenario	Review SQL statements before a project is released.
Procedure	In the top navigation bar of the DMS console, choose **Optimization > SQL Review** and submit the required SQL script that is used to review SQL statements.
Limit	• The operator must have the query permission on the destination database. • The destination database must be a MySQL database. • The control mode of the related instance must be security collaboration.

13.5.8. SQLConsole

Category	Description
Feature	This feature allows you to perform the required operations on the destination database, for example, query or modify the destination database. This feature provides shortcuts to the following quick actions: ● Execute quick queries by opening tables. ● Edit, update, or export query results. ● Open a command window. ● Perform the required operations on tables, for example, rename, edit, delete, modify, or optimize tables.
Scenario	Query or analyze data.
Procedure	Write an SQL statement for the destination database and execute the SQL statement.
Limit	The operator must have the query permission on the destination database or table.

13.5.9. Cross-instance queries

Category	Description
Feature	This feature allows you to perform real-time join queries across online heterogeneous data sources that are deployed in different environments.
Scenario	Perform quick queries and analysis on data across multiple instances.
Procedure	In the top navigation bar of the DMS console, choose **System Management > Instance**, and enable the cross-database query feature. Then, create database links. In the **Cross-database Query** dialog box, submit an executable SQL statement.
Limit	● The operator must have the query permission to the destination database or table. ● The control mode of the related instance must be security collaboration.

13.5.10. System management

13.5.10.1. Instance management

Category	Description
Feature	This feature allows you to centrally manage Alibaba Cloud Apsara Stack database instances. The following databases are supported: ● Relational databases: MySQL, SQL Server, PostgreSQL, PolarDB-X (previously called DRDS), Oracle, and ApsaraDB for OceanBase. ● NoSQL databases: ApsaraDB for Redis and ApsaraDB for MongoDB. ● Analytical databases: AnalyticDB for MySQL and AnalyticDB for PostgreSQL.

Category	Description
Scenario	Manage instances.
Procedure	Create a database account and password that are specific to DMS for the database instance to be managed. In the top navigation bar of the DMS console, choose **System Management > Instance** and add the database instance. **Note** If the control mode is security collaboration, you must associate the required security rule. However, if the control mode is flexible management, you cannot associate security rules.
Limit	The operator must assume the administrator or DBA role.

13.5.10.2. User management

Category	Description
Feature	This feature allows you to centrally manage users in DMS. You can use this feature to manage Apsara Stack tenant accounts, RAM users, and Alibaba Cloud accounts.
Scenario	Manage DMS users.
Procedure	In the top navigation bar of the DMS console, choose **System Management > User** and manage DMS users.
Limit	The operator must assume the administrator role.

13.5.10.3. Task management

Category	Description
Feature	This feature allows you to centrally manage tasks in DMS.
Scenario	Create tasks, stop running tasks, delete tasks, restart failed tasks, and view historical tasks.
Procedure	In the top navigation bar of the DMS console, choose **System Management > Task**. On the page that appears, manage tasks that are created by different tickets, for example, create or modify a task.
Limit	The operator must assume the administrator or DBA role.

13.5.10.4. Security rules

Category	Description
Features	This feature allows you to perform fine-grained instance-level access control on operations, development specifications, development processes, and approval processes.
Scenario	Create instance-level development specifications, development processes, and approval processes.
Procedure	In the top navigation bar of the DMS console, choose **System Management > Security Rules** and configure the required information. The information includes the approval process and related approvers.
Limit	● The operator must assume the administrator or DBA role. ● The control mode of the related instance must be secure collaboration.

13.5.10.5. Approval processes

Category	Description
Feature	This feature allows you to specify approval processes, the approval nodes of each approval process, and the approvers of each approval node.
Scenario	Create a custom approval process.
Procedure	In the top navigation bar of the DMS console, choose **System Management > Security > Approval Processes**, create an approval process, and then specify the required information, such as approvers.
Limit	● The operator must assume the administrator or DBA role. ● The control mode of the related instance must be security collaboration.

13.5.10.6. Operational logs

Category	Description
Feature	This feature allows you to view the logs that record the behaviors of all users.
Scenario	Audit the operations of a database, table, or operator in a specified period of time.
Procedure	In the top navigation bar of the DMS console, choose **System Management > Operation audit** and query the required information.
Limit	The operator must assume the administrator or DBA role.

13.5.10.7. IP whitelisting

Category	Description
Feature	This feature allows you to control access from source IP addresses to Data Management (DMS).
Scenario	Limit the range of IP addresses that can access DMS.
Procedure	In the top navigation bar of the DMS console, choose **System Management > Security > Access IP Whitelists** and specify whether to enable the feature. If this feature is enabled, you can specify CIDR blocks in the IP whitelist.
Limit	The operator must assume the administrator or DBA role.

13.5.10.8. Sensitive data management

Category	Description
Feature	This feature allows you to centrally manage fields that are labeled as sensitive or confidential. You can specify de-identification algorithms for these fields, for example, mask some specified positions, or replace some specified characters.
Scenario	Manage de-identification algorithms, for example, show only the last four digits of a mobile phone number or ID card.
Procedure	In the top navigation bar of the DMS console, choose **System Management > Security > Sensitive Data** and specify the required control method and threshold.
Limit	• The operator must assume the security administrator, database administrator (DBA), or administrator role. • The destination database must be a relational or analytic database. • An instance is in Secure Collaboration mode.

13.5.10.9. Row-level sensitive data management

Category	Description
Feature	This feature allows you to grant row-specific permissions to users. This applies to rows that are included in the same table.
Scenario	View user-specific data. For example, if you need to view the details of a chain enterprise, you can view the details of the region only for which you are responsible and cannot view the details of other regions.
Procedure	In the top navigation bar of the DMS console, choose **System Management > Security > Sensitive Data**. Then, click the **Row Level Security** tab on the left side of the page and set the required resources to be managed. These resources include the databases, tables, fields, and values of fields.

Category	Description
Limit	- The operator must assume the security administrator, database administrator (DBA), or administrator role. - The destination database must be a relational or analytic database. - An instance is in Secure Collaboration mode.

13.5.10.10. Configuration management

Category	Description
Feature	This feature allows you to manage global parameters that apply to Data Management. For example, you can specify a base threshold that includes the maximum number of rows that can be queried each day, or the maximum number of queries that can be issued each day.
Scenario	Create global control policies.
Procedure	In the top navigation bar of the DMS console, choose **System Management > Configuration** and specify the required control method and threshold.
Limit	The operator must assume the administrator role.

14.Server Load Balancer (SLB)

14.1. 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.

SLB consists of three components:

- SLB instances

  An SLB instance is a key load-balancing component in SLB. It receives traffic and distributes traffic to backend servers. To get started with SLB, you must create an SLB instance and add at least one listener and two ECS instances to the SLB instance.

- Listeners

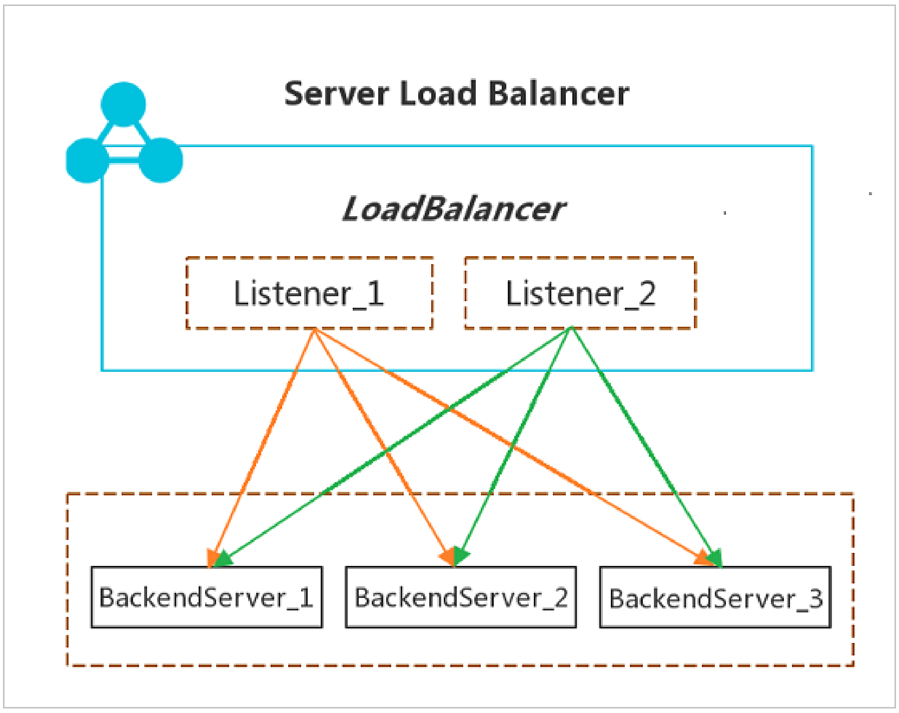
  A listener checks for connection requests from clients, forwards requests to backend servers, and performs health checks on backend servers.

  You can create listeners for Layer-4 (TCP and UDP) or Layer-7 (HTTP and HTTPS) load balancing. For Layer-7 listeners, you can create domain- and URL- based forwarding rules.

- Backend servers

  ECS instances are used as backend servers in SLB to receive and process distributed requests. You can create server groups to categorize your ECS instances in different ways, for example, by use case or by application.

  After an SLB instance receives client requests, the listeners of the SLB instance forward the requests to corresponding backend ECS instances based on the configured forwarding rules, as shown in the following figure.



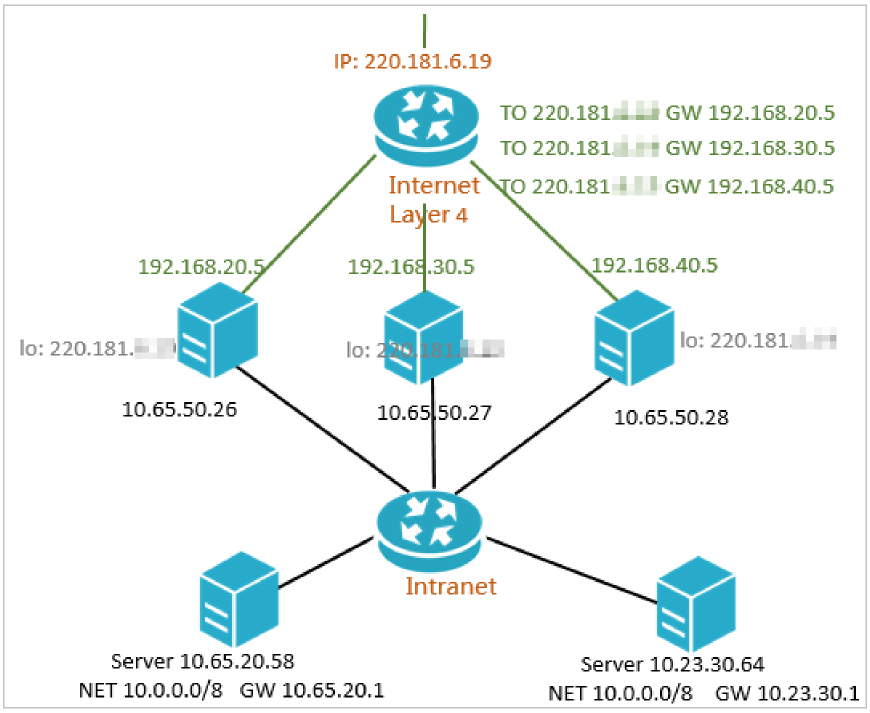
14.2. Architecture

This topic describes the SLB architecture. SLB instances are deployed in clusters to synchronize sessions and protect backend servers from single points of failures (SPOFs), improving redundancy and ensuring service stability.

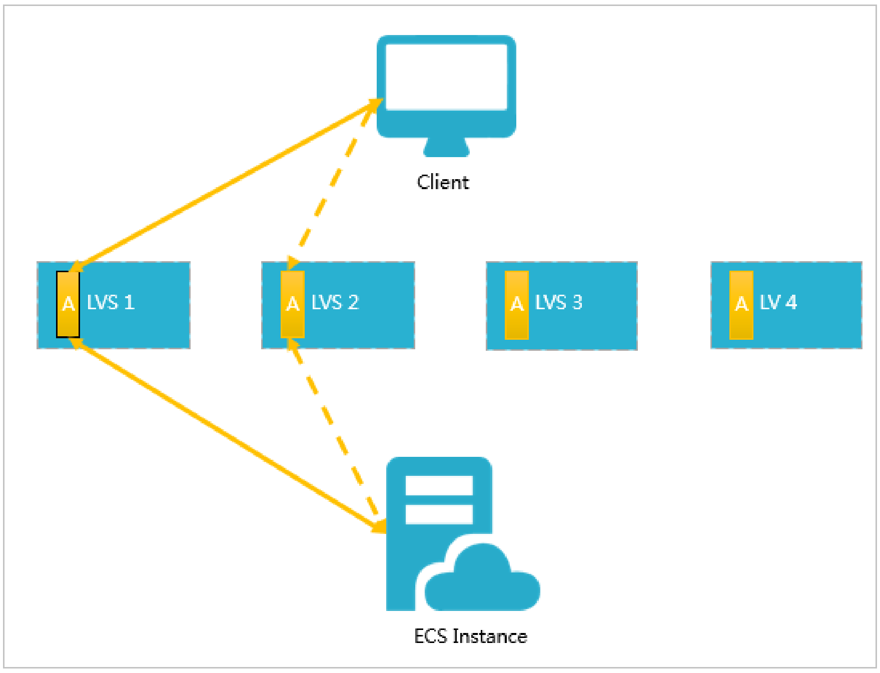
Apsara Stack provides Layer-4 (TCP and UDP) and Layer-7 (HTTP and HTTPS) load-balancing services.

- Layer-4 SLB combines the open-source Linux Virtual Server (LVS) with Keepalived to balance loads, and implements customized optimizations to meet cloud computing requirements.
- Layer-7 SLB uses Tengine to balance loads. Tengine is a web server project launched by Taobao. Based on NGINX, Tengine has a wide range of advanced features optimized for high-traffic websites.

Layer-4 SLB runs in a cluster of LVS machines, as shown in the following figure. This cluster deployment model strengthens the availability, stability, and scalability of the load balancing service in abnormal cases.



In an LVS cluster, each machine uses multicast packets to synchronize sessions with the other machines. Session A established on LVS1 is synchronized to other LVS machines after the client transfers three data packets to the server, as shown in the following figure. Solid lines indicate the current active connections, while dotted lines indicate that the session requests will be sent to other normally working machines if LVS1 fails or is being maintained. In this way, you can perform hot updates, machine maintenance, and cluster maintenance without affecting business applications.



14.3. Function principles

This topic describes the working principles of SLB. SLB distributes inbound network traffic across multiple ECS instances that act as backend servers based on forwarding rules. You can use SLB to improve the responsiveness and availability of your applications.

After you add ECS instances to an SLB instance, SLB uses virtual IP addresses (VIPs) to virtualize the 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 SPOFs. This enhances the resilience of your applications. You can also use SLB to defend your applications against distributed denial of service (DDoS) attacks

14.4. Benefits

14.4.1. LVS in Layer-4 SLB

This topic describes the customized technical improvements on LVS.

Drawbacks of LVS

LVS is an open-source project established by Dr. Zhang Wensong in May 1998. It is now the world's most popular Layer-4 load-balancing software for Linux kernel-based operating systems. LVS is implemented as a kernel module named IP Virtual Server (IPVS) in the netfilter framework, which is similar to iptables. LVS is hooked into LOCAL_IN and FORWARD.

In a large-scale cloud computing network, LVS has the following drawbacks:

- Drawback 1: LVS supports three packet forwarding modes: NAT, DR, and TUNNEL. When these forwarding modes are deployed in a network with multiple VLANs, the network topology becomes complex and incurs high O&M costs.
- Drawback 2: Compared with commercial load-balancing devices such as F5, LVS lacks defense against

DDoS attacks.

- Drawback 3: LVS uses PC servers and the Virtual Router Redundancy Protocol (VRRP) of Keepalived to deploy primary and secondary nodes for high availability. Therefore, its performance cannot be extended.
- Drawback 4: The configurations and health check performance of the Keepalived program are insufficient.

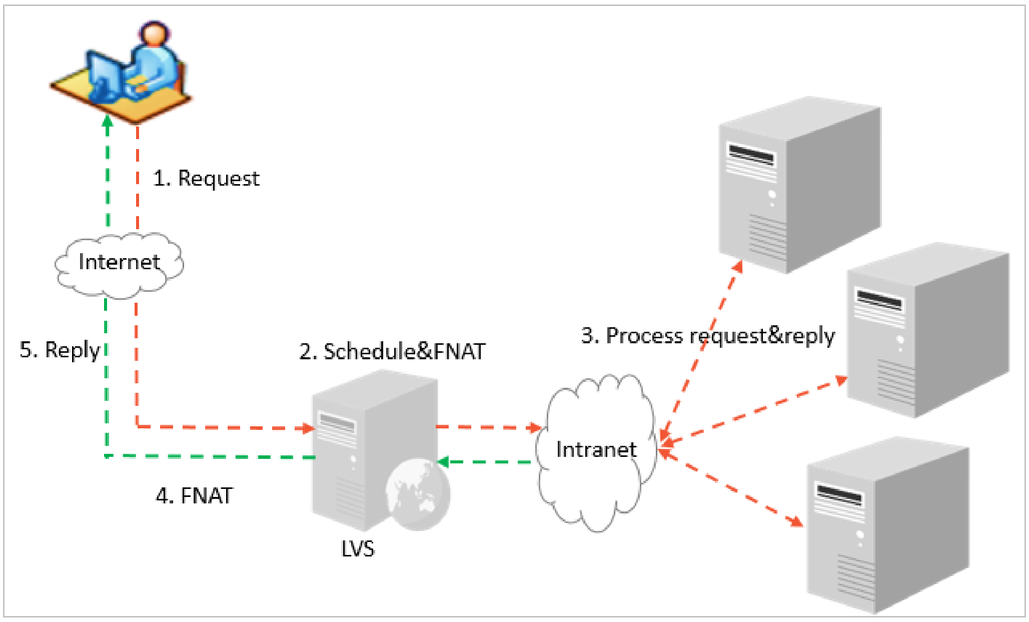
LVS customized features

To solve these problems, Alibaba Cloud added the following customized features to LVS. For more information about Ali-LVS, visit https://github.com/alibaba/LVS.

- Customization 1: FULLNAT, a new forwarding mode that enables inter-VLAN communication between LVS load balancers and backend servers.
- Customization 2: Defense modules such as SYNPROXY against TCP flag-targeted DDoS attacks.
- Customization 3: Support for LVS cluster deployment.
- Customization 4: Improved Keepalived performance.

FULLNAT technology

- Principles: The module introduces local IP addresses (internal IP addresses). IPVS translates CIP (client IP address)-VIP to LIP (local IP address)-RIP (real IP address), in which both LIP and RIP are internal IP addresses. This means that the load balancers and backend servers can communicate across VLANs.
- All inbound and outbound data flows traverse LVS. 10-GE Network Interface Cards (NICs) are used to ensure adequate bandwidth.
- FULLNAT supports only TCP.

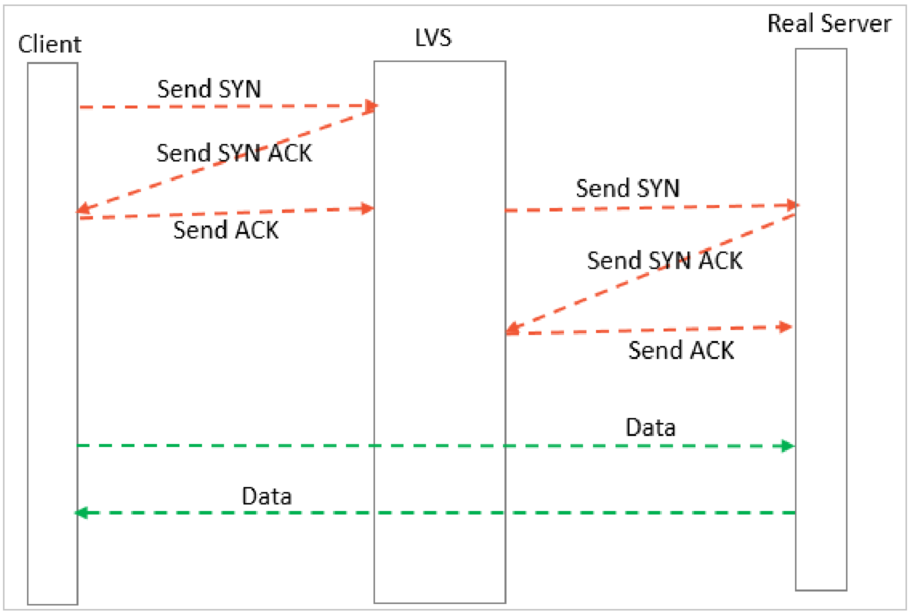


SYNPROXY technology

LVS uses the SYNPROXY module to defend against TCP flag-targeted attacks and SYN flood attacks. Based on the principle of SYN cookies in the Linux TCP protocol stack, LVS acts as a proxy for TCP three-way handshakes.

The process consists of the following steps:

1. A client sends an SYN packet to LVS.
2. LVS constructs an SYN-ACK packet with a unique sequence number and sends this packet to the client. The client returns an ACK response to LVS.
3. LVS verifies the validity of the sequence number in the ACK packet. If the sequence number is valid, LVS establishes a three-way handshake with the backend server.



To defend against ACK, FIN, and RST flood attacks, LVS checks the connection table and discards all requests for connections that are not defined in the table.

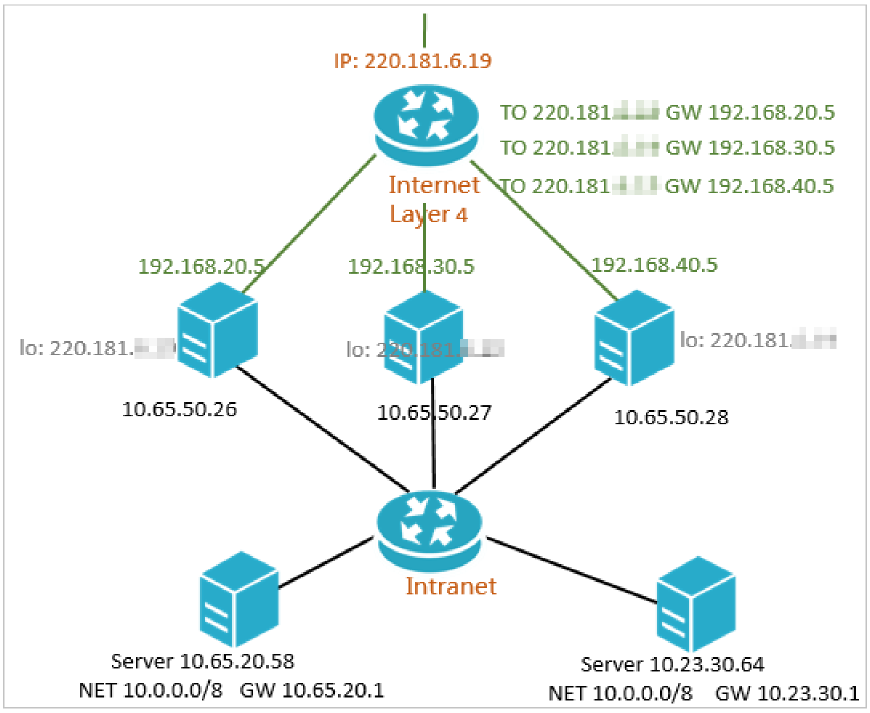
Cluster deployment

An LVS cluster communicates with uplink switches over Open Shortest Path First (OSPF). The uplink switches use equal-cost multi-path (ECMP) routes to distribute traffic to the LVS cluster. Then, the LVS cluster forwards the traffic to your servers.

The cluster deployment model ensures the stability of Layer-4 SLB with the following features:

- Robustness: LVS and uplink switches use OSPF as the heartbeat protocol. A VIP is configured on all LVS nodes in the cluster. The switches can locate the failure of any LVS node and remove it from the ECMP route list.
- Scalability: You can scale out an LVS cluster if traffic from a VIP exceeds the cluster capacity.

Cluster deployment



Keepalived optimization

Improvements made to Keepalived include:

- Change the asynchronous network model from select to epoll.
- Optimize the reloading process.

Features of Layer-4 SLB

In conclusion, Layer-4 SLB has the following features:

- High availability: The LVS cluster ensures redundancy and prevents SPOFs.
- Security: Together with Apsara Stack Security, LVS provides quasi-real-time defense.
- Health check: Health checks are performed on backend ECS instances to automatically remove unhealthy ones from the server pool until they restore.

14.4.2. Tengine in Layer-7 SLB

Tengine is a Web server project launched by Alibaba. Based on NGINX, Tengine has a wide range of advanced features enabled for high-traffic websites. NGINX is one of the most popular open-source Layer-7 load-balancing software.

For more information about Tengine, visit http://tengine.taobao.org/.

Customized features

Tengine is customized for cloud computing scenarios:

- Inherits all features of NGINX 1.4.6 and is fully compatible with NGINX configurations.
- Supports the dynamic shared object (DSO) module. This means you do not need to recompile Tengine to add a module.
- Provides enhanced load balancing capabilities, including a consistent hash module and a session persistence module. It can also actively perform health checks on back-end servers and automatically

enable or disable servers based on their status.

- Monitors system loads and resource usage to protect the system.
- Provides error messages to help locate abnormal servers.
- Provides an enhanced protection module (by limiting the access speed).

Features of Layer-7 SLB combined with Tengine

Layer-7 Server Load Balancer (SLB) is based on Tengine, and has the following features:

- High availability: The Tengine cluster ensures redundancy and prevents single points of failure (SPOFs).
- Security: Tengine provides multi-dimensional protection against CC attacks.
- Health check: Tengine performs health check on back-end ECS instances and automatically isolates abnormal instances until they recover.
- Supports Layer-7 session persistence.
- Supports consistent hash scheduling.

15.Virtual Private Cloud (VPC)

15.1. What is a VPC?

A virtual private cloud (VPC) is a logically isolated virtual network.

Background information

The continuous development of cloud computing technologies leads to increasing virtual network requirements such as scalability, security, reliability, privacy, and performance. This scenario has hastened the birth of a variety of network virtualization technologies.

Earlier solutions combined virtual and physical networks to form a flat network architecture, such as large layer-2 networks. As the scale of virtual networks grew, earlier solutions faced more serious problems. A few notable problems include ARP spoofing, broadcast storms, and host scanning. Various network isolation technologies emerged to resolve these problems by completely isolating the physical networks from the virtual networks. One of the technologies utilized VLAN to isolate users, but due to VLAN limitations, it could only support up to 4096 nodes. It is insufficient to support the huge amount of users in the cloud.

Benefits

A VPC has the following benefits:

- **High security**

  Each VPC has an exclusive and unique tunnel ID, and a tunnel ID corresponds to only one VPC. VPCs are isolated by tunnel IDs.

- **Ease of use**

  You can quickly and easily create and manage a VPC in the VPC console. When you create a VPC, the system automatically provisions a VRouter and a route table for your VPC.

- **High scalability**

  A VPC can be partitioned into multiple subnets to deploy different services. Additionally, you can connect a VPC to an on-premises data center or another VPC to extend the network architecture.

Scenarios

VPCs allow you to flexibly customize the network configuration in the following scenarios:

- **Host Internet-facing applications**

  You can host Internet-facing applications in VPCs and enforce access limits with security group rules and whitelists. VPCs enable you to launch web servers in a public subnet but run your databases in private subnets for isolation and security purposes.

- **Host applications that require access to the Internet**

  By hosting an application in a subnet of a VPC, you can allow this application to receive Internet traffic by using a NAT gateway that provides source network address translation (SNAT). An SNAT rule allows outbound connectivity from the subnet to the Internet without exposing the private IP address of your instance. Furthermore, you can change the public IP address used in an SNAT mapping as needed to prevent targeted attacks.

- **Implement zone-disaster recovery**

Multiple VSwitches can be created in a VPC as subnets. Since VSwitches within a VPC can communicate with each other, they can be used to host your resources in different zones to implement zone-disaster recovery.

- **Isolate business units**

  You can utilize the logical boundaries between VPCs to isolate business units, such as production and test environments. When these business units need to communicate with each other, you can create a peering connection between the VPCs they reside to route traffic between them.

- **Extend your on-premises IT infrastructure**

  To expand the capacity of the existing infrastructure, you can establish a connection between your on-premises data center and a VPC. Moreover, your IT resources can be seamlessly migrated to the cloud without changing how users access these applications.

15.2. Benefits

A VPC is a logically isolated virtual network based on the mainstream tunneling technology.

Each VPC is identified by a unique tunnel ID. Different VPCs are isolated by tunnel IDs:

- Similar to traditional networks, VPCs can also be divided into subnets. ECS instances in the same subnet use the same VSwitch to communicate with each other, whereas ECS instances in different subnets use VRouters to communicate with each other.
- VPCs are completely isolated from each other and can only be interconnected by mapping an external IP address (EIP or NAT IP address).
- The IP packets of an ECS instance are encapsulated by using the tunneling technology. Therefore, information about the data link layer (the MAC address) of the ECS instance is not transferred to the physical network. This way, ECS instances in different VPCs are isolated at Layer 2.
- ECS instances in VPCs use security groups as firewalls to control the traffic to and from ECS instances. This way, ECS instances in different VPCs are isolated at Layer 3.

15.3. Architecture

A VPC is a private network logically isolated from other virtual networks.

Network architecture

Each VPC consists of a private Classless Inter-Domain Routing (CIDR) block, a VRouter, and at least a VSwitch.

- **CIDR blocks**

  A CIDR block is a private IP address range in a VPC. The IP addresses of all cloud resources deployed in the VPC are within the specified CIDR block. When creating a VPC or a VSwitch, you must specify the private IP address range in the form of a CIDR block.

  You can use any of the following standard CIDR blocks and their subnets as the IP address range of the VPC.

CIDR block	Number of available private IP addresses (system reserved ones excluded)
192.168.0.0/16	65,532

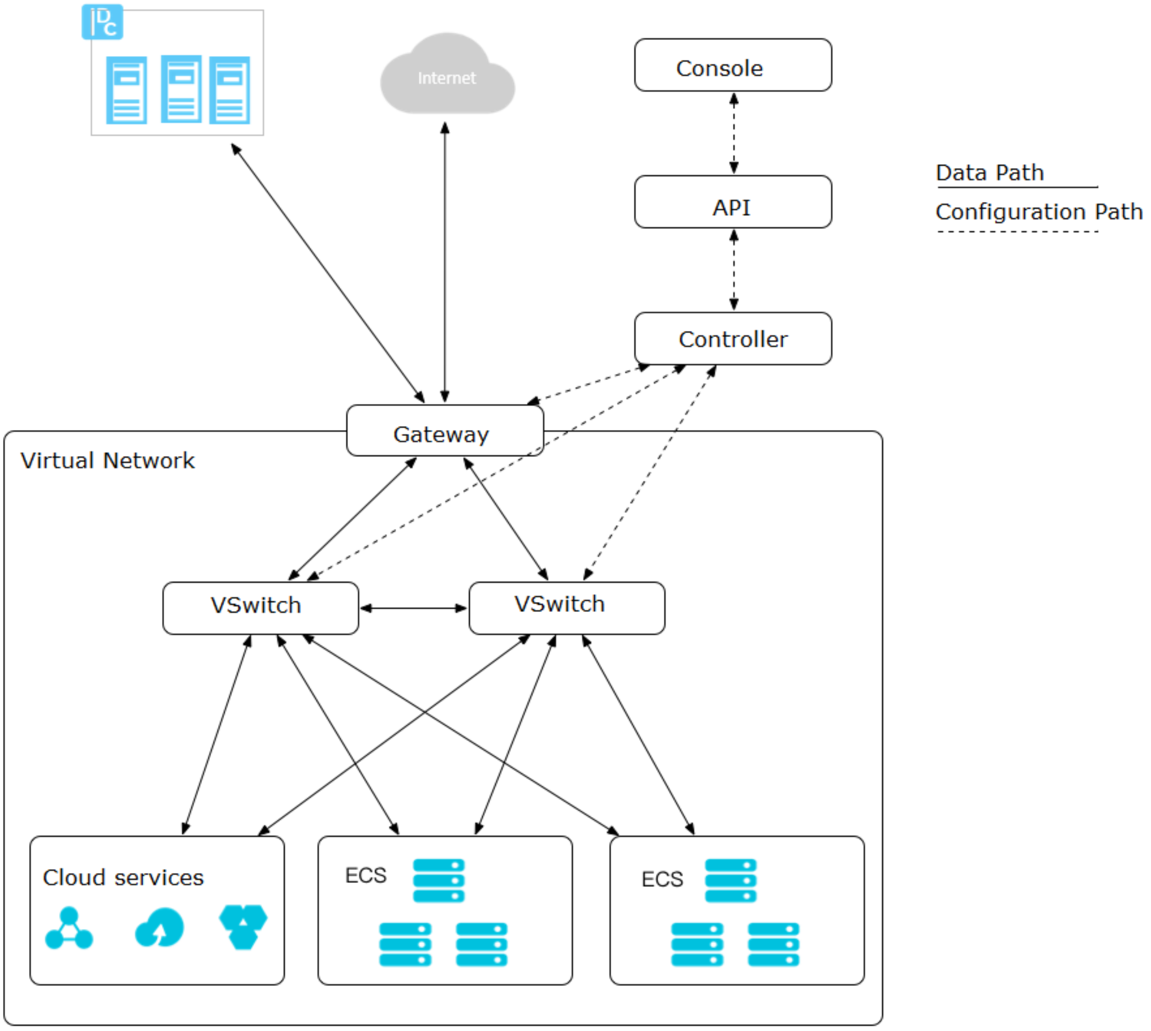
CIDR block	Number of available private IP addresses (system reserved ones excluded)
172.16.0.0/12	1,048,572
10.0.0.0/8	16,777,212

- **VRouters**

  A VRouter is the hub of a VPC. A VRouter is also an important component of a VPC. The VRouter connects the VSwitches in a VPC and serves as the gateway connecting the VPC with other networks. After you create a VPC, the system automatically creates a VRouter, which is associated with a routing table.

- **Switches**

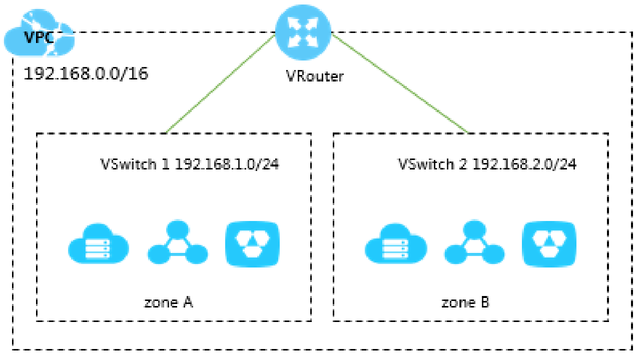
  A VSwitch is a basic network device in a VPC and is used to connect different cloud product instances. After creating a VPC, you can further divide the VPC into one or more subnets by creating VSwitches. The VSwitches within a VPC are interconnected. You can deploy applications in VSwitches of different zones to improve the service availability.



System architecture

The VPC architecture contains the VSwitches, gateway, and controller. The VSwitches and gateway form the key data path. Controllers use the protocol developed by Alibaba Cloud to forward the forwarding table to the gateway and VSwitches, completing the key configuration path. In the overall architecture, the configuration path and data path are separated from each other. VSwitches are distributed nodes. The gateway and controller are deployed in clusters. Multiple data centers are built for backup and disaster recovery. Redundant links are provided for disaster recovery. This deployment mode improves the overall availability of the VPC.

VPC architecture



15.4. Features

A VPC is a logically isolated virtual network based on the mainstream tunneling technology.

Each VPC is identified by a unique tunnel ID. A unique tunnel ID is generated when tunnel encapsulation is performed on each data packet transmitted between the ECS instances within a VPC. Then, the data packet is transmitted over the physical network. ECS instances in different VPCs cannot communicate with each other. They have different tunnel IDs and therefore are on different routing planes.

Alibaba Cloud developed technologies such as the VSwitch, Software Defined Network (SDN), and hardware gateway based on the tunneling technology. These technologies serve as the basis for VPCs.

16.Apsara Stack Security

16.1. 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.
	Cloud Firewall	Allows you to centrally manage access control policies for traffic transferred within your business system (east-west) and between the Internet and your business system (north-south).
Data security	Sensitive Data Discovery and Protection (SDDP)	Prevents data leaks and helps your business system meet compliance requirements.

Security domain	Service name	Description
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.

16.2. Advantages

Since the enforcement of China Internet Security Law, Regulations on Critical Information Infrastructure Security Protection and Cloud Security Classified Protection Standard 2.0 have been published. As a result, private cloud platforms must pass the classified protection evaluation to ensure the security of cloud systems. Increasing security threats such as attacker intrusions and ransomware have led to the rising needs for security issue detection and prevention.

At the network perimeter of Apsara Stack, Apsara Stack Security uses a traffic security monitoring system to detect and block network-layer attacks in real time. It detects and removes Trojans and malicious files on servers to prevent attackers from exploiting the servers. In addition, Apsara Stack Security can block brute-force attacks and send alerts on unusual logons. This prevents attackers from stealing or destroying business data after logging on the system with weak passwords.

In-depth defense system

Apsara Stack Security comprises multiple functional modules. These modules work together to provide in-depth defense on the Apsara Stack network perimeter, within the Apsara Stack network, and on the Elastic Compute Service (ECS) instances in Apsara Stack. To help you manage security risks of Apsara Stack in a centralized manner and in real time, Apsara Stack Security provides a unified security management system. This system allows you to manage the security policies in all security protection modules and perform association analysis on the logs.

The security protection modules provided by Apsara Stack Security cover network security, server security, application security, and threat analysis. Based on a management center that can integrate the security information from all modules, Apsara Stack Security can accurately detect and block attacks. In this way, Apsara Stack Security protects your business systems in the cloud against intrusions.

Security solutions completely integrated with the cloud platform

Apsara Stack Security is a product born from ten years of protection experience. After a decade of experience in providing security operations services for the internal businesses of Alibaba Group and six years of safeguarding the Alibaba Cloud security operations, Alibaba has obtained considerable security research achievements, security data, and security operations methods, and has built a professional cloud security team. Apsara Stack Security brings together the rich experience of these experts to develop the sophisticated systems that provide enhanced security for cloud computing platforms. This product can protect the cloud platform, cloud network environments, and cloud business systems of Apsara Stack users.

The components of Apsara Stack Security are software-defined, with a full hardware compatibility. With these components, you can implement elastic cloud computing services based on quick deployment, expansion, and implementation. The protection modules on the cloud network perimeter or in the cloud network adopt the bypass architecture, which completely fits the cloud businesses and has the minimal adverse impacts on the cloud businesses. The protection modules running on the ECS instances are all virtualized to fit the flexibility of the ECS instances.

User security situation awareness

The cloud platform provides services for users. In Apsara Stack Security console, a user can view the security protection data, generate security reports, and enable SMS and email alerts by configuring external resources.

Security capability output

Apsara Stack Security has accumulated a large number of protection policies over the last several years. The service has protected millions of users from hundreds of thousands of attacks every day. This has generated a large amount of security protection data. Apsara Stack Security analyzes over 10 TB of this data every day. The analysis results are used to enhance the fundamental security capabilities, such as the malicious IP library, malicious activity library, malicious sample library, and vulnerability library. These capabilities are applied in the protection modules of Apsara Stack Security to enhance your business security.

16.3. Architecture

Apsara Stack Security consists of Apsara Stack Security Standard Edition and optional security services.

Apsara Stack Security Standard Edition

- Threat Detection Service (TDS)

  This module collects network traffic and server information and detects possible vulnerability exploits, intrusions, and virus attacks through machine learning and data modeling. This module also provides you with up-to-date information about ongoing attacks to help you monitor the security status of your business.

- Network Traffic Monitoring System

  This module is deployed on the network perimeter of Apsara Stack. This module allows you to inspect and analyze each inbound or outbound packet of an Apsara Stack network through traffic mirroring. The analysis results are used by other Apsara Stack Security modules.

- Asset Vulnerability Monitoring

  This module analyzes known assets based on the built-in asset learning model to identify asset sources and help enterprises automatically detect unknown assets. This module also detects asset vulnerabilities to help enterprises identify unknown security risks in a timely manner.

- Server Security

  This module collects information and performs detection by deploying clients on physical servers. This module monitors the security status of all physical servers in the Apsara Stack environment in real time, and provides a variety of features to help you detect security risks on physical servers in a timely manner. The features include Overview, Servers, Intrusion Detection, Server Fingerprints, and Log Retrieval.

- Server Guard

  This module protects Elastic Compute Service (ECS) instances by providing security features such as vulnerability management, baseline check, intrusion detection, and asset management. To do this, the module performs operations such as log monitoring, file analysis, and signature scanning.

- Web Application Firewall (WAF)

This module protects web applications against common web attacks reported by Open Web Application Security Project (OWASP), such as Structured Query Language (SQL) injections, cross-site scripting (XSS), exploitation of vulnerabilities in web server plug-ins, trojan uploads, and unauthorized access. This module also blocks a large number of malicious requests to avoid data leaks and ensure both the security and availability of your websites.

Apsara Stack Security Standard Edition also provides on-premises security services. These services help you better use the features of Apsara Stack services such as Apsara Stack Security to ensure the security of your applications.

On-premises security services include pre-release security assessment, access control policy management, Apsara Stack Security configuration, periodic security check, routine security inspection, and urgent event handling. These services cover the entire lifecycle of your business in Apsara Stack and help you create a security operations system. This system enhances the security of your application systems and ensures both the security and stability of your business.

Optional security services

You can also choose the following service modules to enhance your system security.

- Anti-DDoS Service

  This module detects and blocks distributed denial of service (DDoS) attacks.

- Cloud Firewall

  This module sorts and isolates different types of business based on visualized business data to implement access control over east-west traffic in Apsara Stack.

- Sensitive Data Discovery and Protection (SDDP)

  This module uses big data analytics capabilities and artificial intelligence (AI) technologies of Alibaba Cloud to detect and classify sensitive data based on your business requirements. This module can also mask sensitive data both in transit and at rest, monitor dataflows, and detect abnormal activities. This module provides visible, controllable, and industry-compliant security protection for your sensitive data by using precise detection and analysis.

16.4. Features

16.4.1. Apsara Stack Security Standard Edition

16.4.1.1. Threat Detection Service

Threat Detection Service (TDS) is a system developed by the Alibaba Cloud security team for analyzing big data security.

This system analyzes server and network traffic to detect possible threats or attacks by using machine learning and data modeling. It identifies vulnerable exploits and potential virus attacks, and provides you with up-to-date information about ongoing attacks to help you monitor the security status of your businesses.

Features

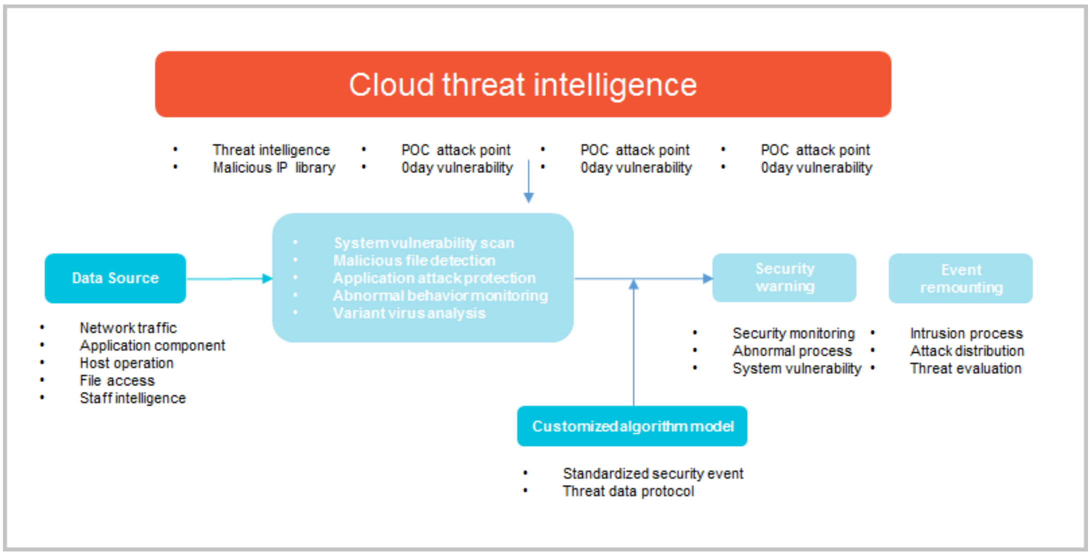
The following table describes features of TDS.

Feature	Description
Overview	Provides a comprehensive security overview with statistics on security score, asset status, unhandled alerts, and handled alerts.
Security Dashboard	Displays the security data on dashboards, including assets, vulnerabilities, baselines, attack sources, and attack distribution.
Security Alerts	Allows you to view and handle security events, including suspicious process, webshells, unusual logons, sensitive file tampering, malicious processes, suspicious network connections, and web application threat detection.
Attack Analysis	Displays the attack trends and attack type distribution in the last 7 days and 30 days. Displays the attack information such as the attack time, attack source, attacked assets, number of attacks, risk level, and attack type.
Cloud Service Check	Checks the security configurations of cloud services from the aspects of network access control and data security. It supports periodic checks that run automatically and manual checks. You can verify the check results or configure whitelist policies for the check results.
Application Whitelists	Allows you to add servers to a whitelist based on intelligent learning and identifies programs as trusted, suspicious, or malicious based on the whitelist. Unauthorized processes will be terminated.
Assets	● Server: displays the security statuses for servers. You can view the numbers of all servers, risky servers, unprotected servers, inactive servers, and new servers. ● Cloud Product: provides security status information for cloud services and supports SLB and NAT.
Security Reports	Allows you to query reports. For example, you can retrieve historical reports by report name.

How it works

The following figure shows how TDS works.

TDS working principle



- Big data security analysis platform
  - Network: TDS uses HTTP requests and responses collected by the traffic security monitoring module to create HTTP logs. It uses big data models to analyze the logs and discover security events and threats.
  - Server: TDS uses the rule engine to analyze server process data collected by Server Guard and discover security events and threats.
- Security event display
  - Security events reported by Server Guard
  - Server security events discovered in server process analysis conducted by the rule engine
  - Network security events discovered in HTTP log analysis conducted by big data models

Benefits

TDS provides the following benefits:

- Big data-based threat analysis

  TDS analyzes and computes petabyte-level big data. It collects all security data and threat information from the entire network. It also uses the machine learning technology to create comprehensive, intelligent security threat models that can be used in business scenarios with millions of users.

  This service focuses on the security trends and new threats that are faced by users of cloud computing services in data centers, such as targeted web application attacks and system brute-force attacks. It defends your systems against diverse threats.
- Dashboard

  To facilitate security decision making on Apsara Stack, TDS displays the results of big data threat analysis in graphs by using Internet visualization technologies.

16.4.1.2. Traffic Security Monitoring

The Traffic Security Monitoring module is an Apsara Stack Security service that can detect attacks within milliseconds.

By performing in-depth analysis on the traffic packets mirrored from the Apsara Stack network ingress, this module can detect various attacks and unusual activities in real time and coordinate with other protection modules to implement defenses. The Traffic Security Monitoring module provides a wealth of information and basic data support for the entire Apsara Stack Security defense system.

Features

The following table describes the features that the Traffic Security Monitoring module provides.

Feature	Description
Traffic data collection and analysis	Uses a bypass in traffic mirroring mode to collect inbound and outbound traffic that passes through the interconnection switch (ISW) and generates a traffic diagram.
Unusual traffic detection	Uses a bypass in traffic mirroring mode to detect the unusual traffic that has exceeded the scrubbing threshold and reroutes the traffic to the DDoS Traffic Scrubbing module. The traffic rate (Unit: Mbit/s), packet rate (Unit: PPS), HTTP request rate (Unit: QPS), or number of new connections can be set as the threshold.
Malicious server identification	Detects attacks launched by internal servers to identify controlled malicious servers.
Web application protection	Uses a bypass to block common attacks on Web applications at the network layer based on default Web attack detection rules. The attacks that can be blocked include Structured Query Language (SQL) injections, code and command execution, Trojan scripts, file inclusion attacks, and exploitation of upload vulnerabilities and common content management system (CMS) vulnerabilities.
Suspicious TCP connection blocking	Uses a bypass to send TCP RST packets to the server and the client to block layer-4 TCP connections.
Network log recording	Records UDP and TCP traffic logs and the Request and Response logs of HTTP queries. Threat Detection Service (TDS) uses these logs for big data analysis.

How it works

The Traffic Security Monitoring module collects data, processes the data, and then generates data processing results. It uses sockets to exchange data.

- Collection: The module collects traffic data through multiple high-performance PCs with dual-port 10GE network interface controllers (NICs).
- Processing: Traffic from an IP address may pass through multiple collectors. Traffic data must be consolidated to generate usable information.
- Output: The module stores and provides the consolidated traffic data.

16.4.1.3. Cloud Security Scanner

The Cloud Security Scanner module uses AI technologies to help enterprises identify security risks at the earliest opportunity.

This module analyzes known assets based on the built-in asset learning model to identify asset sources and help enterprises automatically detect unknown assets. This module detects asset vulnerabilities to help enterprises identify unknown security risks.

This module provides 24/7 monitoring on assets, asset vulnerabilities, access control lists (ACLs), and security baselines to detect security risks in real time. This module also notifies you of security risks through text messages and emails. This helps you identify security risks at the earliest opportunity.

Features

The Cloud Security Scanner module provides the following features.

Feature	Description
Asset discovery	Accurately identifies asset sources based on known assets and the built-in asset learning model. This feature inspects assets on a regular basis to detect unknown assets and add them to the asset library. The asset learning model identifies the assets of enterprises. For example, the model can accurately identify the change status of host services and host assets, automatically discover the alive status of assets, configure asset inspection tasks, and discover subdomains and multi-level domain names.
Asset management	Allows you to import, delete, group, export, query, and monitor assets and manage asset owners. This feature also uses deep learning technology to split requests into different applications and services by protocol, extract characteristics, and automatically identify applications and services by using the fingerprint model. This feature can also be used to schedule the Server Guard agent that is deployed nearby to monitor the assets of enterprises.
Asset monitoring	Uses HTTP and ping commands to monitor assets, display the details about the availability of an asset, and display basic monitoring information about a monitored website based on custom alert rules.
Vulnerability scanning	Scans your system for basic vulnerabilities, weak passwords, security vulnerabilities, and Common Vulnerabilities and Exposures (CVE). This feature supports automatic vulnerability inspection and baseline check. The vulnerabilities include common web vulnerabilities, the latest high-risk CVE, common CMS vulnerabilities, and O&M security vulnerabilities. This feature scans weak passwords that are used in services such as MySQL, SSH, FTP, and SQL Server.
Vulnerability management	Automatically associates detected vulnerabilities with assets to visualize asset risks and help enterprises detect and manage risks at the earliest opportunity. This feature supports vulnerability management. You can ignore or confirm vulnerabilities, or rescan assets for vulnerabilities. This feature allows you to view vulnerability details, fix vulnerabilities based on suggestions provided by the feature, and share vulnerability details. This helps reinforce the security of enterprise systems.
External risk monitoring	Detects external risks based on features of employee behavior and key enterprise information. This feature identifies and generates alerts for code uploads to GitHub by your employees. This way, you can monitor, handle, and view the details of code leaks, and the risk of code leaks is reduced.

Scenarios

- Security O&M on small-scale networks

In small-scale networks, the Cloud Security Scanner module is deployed in standalone mode to scan networks. You can deploy this module in small-scale networks for security O&M with ease. The module helps check your business system for various security risks.

- Security O&M on medium-scale multi-subnet networks

  Medium-sized enterprises have medium-scale networks. The networks are divided into multiple service subnets in different regions. To protect the networks of all services, you can deploy the Server Guard agent in each subnet and use the Cloud Security Scanner module to manage the agent in a centralized manner.

- Security O&M on large-scale cross-region networks

  Large enterprises have large-scale cross-region networks. The Cloud Security Scanner module is deployed in each region and managed at the headquarters in a centralized manner. To protect the networks of all regions, you can deploy the Server Guard agent in each subnet and use the Cloud Security Scanner module to manage the agent in a centralized manner.

Benefits

- Data visualization

  Collects and processes asset and risk data, and displays assets and system data in a visual display. The Cloud Security Scanner module monitors host information and website information on the network and displays key information in charts. This allows enterprises to monitor the current status of the business and facilitates subsequent business adjustments.

- Flexible, accurate, and fast scanning

  Accelerates scanning by using stateless scanning technology, uses a distributed scanning architecture and task scheduling module to split scanning requests, and uses fingerprint identification technology to achieve precise scans. This reduces redundant scan requests and avoids excessive server loads due to high scanning frequencies.

- Up-to-date large-scale vulnerability libraries and quick response to the latest high-risk vulnerabilities

  Updates high-risk vulnerability plug-ins in real time and reports vulnerabilities that affect business at the earliest opportunity. Supports thousands of independent application vulnerability plug-ins, hundreds of threat intelligence channels, and 2,000 types of vulnerabilities. The Cloud Security Scanner module monitors the security status of enterprise-related vulnerability platforms and social media tools. This module can also detect vulnerabilities at the earliest opportunity because it can obtain threat intelligence from mainstream vulnerability platforms.

- Accurate discovery of IT assets

  Uses the asset learning model to extract characteristics from enterprises and the assets of the enterprises to build enterprise asset models. The models can be used to identify asset sources. The Cloud Security Scanner module analyzes threats based on enterprise characteristics and discovered asset characteristics to obtain enterprise-related security intelligence.

- Integration of various detection capabilities and unified risk analysis

  Detects vulnerabilities in user systems in a comprehensive manner. The Cloud Security Scanner module detects security vulnerabilities and security configuration issues in information systems, security vulnerabilities in application systems, and weak passwords in systems. This module also collects unnecessary accounts, services, and ports that are open, and generates an overall security risk report. This helps security administrators troubleshoot security issues before attackers discover the issues.

Detects system vulnerabilities and web application vulnerabilities, and supports baseline checks. Analyzes vulnerabilities and assesses risks in the network system in a centralized manner, and evaluates overall security status. This way, you are informed of security risks in information systems.

- Quick identification of high-risk assets from large amounts of data by using custom detection items

  Collects system environment information and maps asset details with asset vulnerabilities. If a major vulnerability occurs, an urgent detection service that is pushed by the system is used to list the affected assets. This helps fix the vulnerability at the earliest opportunity and reduce the impact on the assets.

- Deep security detection with a few clicks

  Provides basic risk monitoring and advanced risk monitoring features to meet different business requirements. Advanced risk monitoring performs in-depth security scans on assets, which may generate a large amount of useless data in databases and affect business continuity. By default, the Cloud Security Scanner module uses basic risk monitoring. To perform an in-depth security scan on your website assets, you can enable the advanced risk monitoring feature.

- Centralized security management processes

  Formulates security management processes to control security risks. Vulnerabilities arise in many enterprises even when security processes are in place because the enterprises do not integrate the security processes into the management processes. The Cloud Security Scanner module helps formulate integrated security management processes to execute security processes.

16.4.1.4. Server Guard

Server Guard provides security protection measures such as vulnerability management, baseline check, intrusion detection, and asset management for Elastic Compute Service (ECS) instances by means of log monitoring, file analysis, and feature scanning.

Server Guard uses the client-server model. To protect the security of ECS instances in real time, Server Guard clients work with the Server Guard server to monitor attacks and vulnerabilities at the system layer and the application layer on the ECS instances.

Features

Category	Feature	Description
Overview	Overview	Displays assets, vulnerabilities, exceptions, configuration defects, and events that require attention.

Category	Feature	Description
Servers	Server Fingerprints	Provides the following modules: • Port: checks and displays the listening port information, including the listening port, protocol, process, IP address, and update time. • Software: checks and displays the software installation information on servers, including the software name, software version, software installation directory, and update time. • Process: checks and displays the process information, including the process name, process path, startup parameter, startup time, user, permission, process ID (PID), parent process, and update time. • Account: checks and displays the host account information, including the account name, logon permission, root permission, user group, expiration time, last logon time, and update time. • Scheduled Tasks: checks and displays the scheduled tasks of the host, including the task path, execution command, task cycle, account name, and update time.
Threat Prevention	Baseline Check	Automatically detects configuration risks related to the system, account, database, weak password, and security compliance on your servers, and provides security hardening suggestions. This feature also checks database, system, and middleware assets.
	Vulnerabilities	Detects four types of vulnerabilities: Linux, Windows, Web CMS, and emergency vulnerabilities and provides vulnerability fix solutions. You can verify vulnerability fixes, view vulnerability details, and identify all vulnerabilities at one click.
Intrusion Prevention	Intrusions	Displays the alert information of affected host assets, including the number of alerting servers, the total number of unhandled alerts, and the number of urgent alerts.
	File Tamper Protection	Supports web page tamper-proofing and provides the blacklist and whitelist prevention modes.
	Virus Removal	Detects and removes virus and webshell. The system automatically detects and removes common trojan viruses, ransomware, mining viruses, and DDoS trojans.
Log Retrieval	Log Retrieval	Allows you to query logs for logon, brute-force attack, process snapshot, network connection, listening port snapshot, account snapshot, and process startup.
Server Settings	Client Installation	Allows you to view offline servers. You can install clients for the servers again based on the Client Installation Guide. You can uninstall the Server Guard client from the specified server.
	Protection Mode	Provides business first and protection first modes for different scenarios.

How it works

Server Guard uses the client-server model. The client is installed on ECS instances. The client communicates with the server through a TCP persistent connection and uses HTTP to obtain scripts, rules, and installer packages from the server.

The client can be used in Windows or Linux. It can automatically connect to the server for online updates.

Server Guard supports the following key features:

- **Vulnerability management**: The client collects the ECS instance information, including component information, software versions, file information, and registry information. Then, the client checks whether the information matches the vulnerability detection rules provided by the server. The information that matches the rules will be sent to the server for further analysis. The detected vulnerabilities will be displayed in the Server Guard console. You can fix vulnerabilities in the console or by calling API operations. After receiving the vulnerability patches from the server, the client on the vulnerable ECS instance automatically fixes the vulnerabilities and synchronizes the vulnerability status to the server.
- **Baseline check**: When you manually start a check or a periodic check is triggered, the Server Guard server sends a baseline check request to the client. The client then collects the server information according to the check policy and compares the information with the security baseline. Check items that do not comply with the baseline are labeled as at-risk items and reported to the server.
- **Unusual logon detection**: The client monitors the logon logs of the server system in real time. In a Linux system, the */var/log/secure* and */var/log/auth.log* files are also monitored. All failed and successful logons are recorded. Unusual logons or brute-force attacks will be reported to the server.
- **Webshell detection**: The client uses an Alibaba-developed dynamic webshell detection engine to detect complex webshells. It then restores these webshells to an identifiable status to analyze the hidden webshell activities. This prevents webshells from bypassing the detection due to the use of static detection rules.
- **Suspicious process detection**: The Server Guard server uses a data analysis rules engine to analyze the server process data collected by the client. By doing so, the server can detect suspicious processes such as reverse shells, mining processes, DDoS trojans, worms, viruses, and hacking tools.
- **Log collection**: The client collects logs such as processes logs and network logs.

Scenarios

Server Guard is applicable to server security protection in the following scenarios:

- **Use common software for website building**

  In this scenario, attackers may intrude servers by exploiting vulnerabilities in common software. You can use Server Guard to detect and fix vulnerabilities.

- **Use Web application services**

  Attackers may steal website data through both internal and external web services. You can use Server Guard to prevent attackers from launching attacks or controlling your servers.

16.4.1.5. Server Security

Server Security is a module that deploys the Server Guard agent on physical servers to collect information and detect security risks. This module monitors the security status of physical servers in the Apsara Stack environment and provides a variety of features to help you detect security risks on physical servers in real time. The features include Overview, Servers, Intrusion Detection, Server Fingerprints, and Log Retrieval.

Features

The Server Intrusion Detection module provides the following features:

Feature		Description
Overview	Overview	Displays the server protection status, the number of abnormal servers, the number of unusual logons, the number of websites that are implanted with webshells, the intrusions that are detected on servers during a specific period, and the servers that experience the largest amount of intrusions.
Hosts	Hosts	Supports asset groups and asset tag management.
Intrusion Prevention	Unusual Logons	• Audits all logons and generates alerts on unusual logons. You can configure the usual logon locations. • Generates alerts on logons by using IP addresses that are not in a logon IP address whitelist. You must configure a logon IP address whitelist first. • Generates logon alerts based on IP addresses that are not in a logon IP address whitelist. You must configure approved logon time ranges first. • Generates alerts on logons by using disapproved accounts. You must configure approved logon accounts first. • Detects unusual logon attempts that are used to crack passwords and reports the attempts to Server Guard to block the attempts. This prevents intrusions that may be caused by brute-force attacks.
	Webshell detection and removal	Uses a self-developed webshell detection engine to detect and remove webshells. You can schedule tasks, and configure protection and scan policies to detect webshells in real time. This feature detects webshell files such as PHP and JSP files.
	Suspicious server detection	Detects suspicious activities such as reverse shells, Java processes that run CMD commands, and unusual file downloads that are completed by using Bash.
Server Fingerprints	Listening ports	Collects and displays port listening information. This feature also records changes to ports to check open ports.
	Account information	Collects information about accounts and related permissions and checks privileged accounts for privilege escalation.
	Processes	Collects and displays process snapshots to track normal processes and detect unusual processes.

Feature		Description
	Software version	Checks software installation information. When high-risk vulnerabilities occur, this feature locates affected assets.
Log Retrieval	Log retrieval	• Process startup: records the details of process startup. • Process snapshot: takes and stores a snapshot of full process logs at a specified point in time. • Outbound connection: collects 5-tuple information about external network connections in real time. • System logon: queries logs of SSH and RDP logon processes. • Port listening snapshot: takes and stores a snapshot of all listening ports at a specified point in time. • Account snapshot: takes and stores a snapshot of all accounts at a specified point in time.
Settings	Security settings	• Enables periodic trojan scans for servers. • Specifies the working mode of the Server Guard agent. You can select the business first or protection first mode.

How it works

Server Security works in client-server mode. In this mode, the Server Guard client is installed on each physical server. The client communicates with the server by using a TCP persistent connection and uses HTTP to obtain scripts, rules, and installer packages from the server.

The following list describes the core features of Server Security:

- Unusual logon detection

  The Server Guard client monitors the logon logs of the physical server system in real time. In a Linux system, the */var/log/secure* and */var/log/auth.log* files are monitored. Failed and successful logons are recorded. Unusual logons or brute-force attacks are reported to the server.

- Webshell detection

  The Server Guard client uses an Alibaba-developed dynamic webshell detection engine to detect complex webshells. To analyze hidden webshell activities, the client restores the webshells to an identifiable status. This way, webshells cannot bypass detection when only static detection rules are used.

- Suspicious server detection

  The Server Guard server uses a data analysis rules engine to analyze the server process data collected by the client. The server can detect suspicious processes such as reverse shells, mining processes, DDoS trojans, worms, viruses, and hacking tools.

- Log collection

  The Server Guard client collects logs such as process logs and network logs.

16.4.1.6. WAF

Web Application Firewall (WAF) protects the web applications of cloud users against common web attacks.

Different from traditional web application firewalls, Apsara Stack WAF uses intelligent semantic analysis algorithms to identify web attacks. WAF also integrates a learning model to enhance its analysis capability so that it can meet your daily security protection requirements without relying on traditional rule libraries.

WAF protects the traffic of businesses on HTTP and HTTPS websites. In the WAF console, you can import certificates and private keys to enable end-to-end encryption. This prevents the interception of business data on the links.

WAF not only prevents common web application attacks defined by Open Web Application Security Project (OWASP) but also mitigates HTTP flood attacks. In addition, WAF allows you to customize protection policies based on the businesses of your website to block malicious web requests.

Features

The following table describes the features provided by WAF.

Category	Feature	Description
Detection Overview	Detection Overview	Provides statistics on protection for the last 24 hours and the last 30 days.
	Access Status Monitor	Displays the top 100 access requests in real time.
	Export Detection Report	Allows you to export daily reports, weekly reports, and scheduled task reports.
	Attack Detection Statistics	Provides statistics on attack detection.
Detection Logs	Attack Detection Logs	Provides attack detection logs. The log list displays the processing results, attacked addresses, attack types, attacker IP addresses, and attack time. You can view log details for each attack.
	HTTP Flood Detection Logs	Provides HTTP flood protection logs. The log list displays logs for matched HTTP flood protection rules, including the request URL, the name of the matched rule, and the match time. You can filter logs based on the event generation time and the name of the HTTP flood protection rule.
	System operation log	Provides system operations logs, including usernames, operations, and IP addresses.
	Access Log	Provides access logs, including the access address, destination IP address, source IP address, request method, and response code.
	Protection site management	Allows you to create, delete, modify, enable, and disable function forwarding proxies of a protected site.

Category	Feature	Description
Protection Configuration	Customized Rules	Allows you to create, delete, enable, and disable custom rules. This implements fine-grained HTTP access control for websites.
	Website Protection Policies	• Supports decoding methods, such as URL decoding, JSON parsing, Base64 decoding, hexadecimal conversion, backslash unescape, XML parsing, PHP deserialization, and UTF-7 decoding. • Detects SQL injections, cross-site scripting (XSS), intelligence, cross-site request forgery (CSRF), server-side request forgery (SSRF), Hypertext Preprocessor (PHP) deserialization, Java deserialization, Active Server Pages (ASP) code injections, file inclusion attacks, file upload attacks, PHP code injections, command injections, crawlers, and server responses. • Provides five built-in protection templates, including the template with default protection policies, monitoring mode template, anti-DDoS template, template for financial customers, and template for Internet customers. WAF allows you to customize the decoding algorithms in the templates, enable or disable each attack detection module separately, and configure the detection granularity. WAF also allows you to specify the Block Status Code parameter. • Allows you to enable HTTP response detection and configure the length of the response body in detection rules. • Allows you to configure the length of the request body in detection rules. • Allows you to enable or disable detection timeout settings.
	HTTP Flood Protection	Allows you to configure access frequency control rules for domain names and URLs. This restricts the access frequency of IP addresses or sessions that meet the criteria, or blocks these IP addresses or sessions. Restricts the access frequency of known IP addresses or sessions or blocks these IP addresses or sessions. Supports the HTTP flood protection whitelist function. HTTP flood protection rules are not applicable to IP addresses or sessions in a whitelist.
	SSL Certificate Management	Allows you to upload certificate files and SSL private keys to manage SSL certificates.
System Management	Node status	• Payload Status: displays the CPU utilization and memory usage. • Node Network Status: displays the read throughput and write throughput. • Detection Status: displays the queries per second (QPS) and the average detection time consumed by WAF nodes. • Forward Status: displays the number of new connections per second and the average latency. • Disk Status: displays the disk usage and total disk size.

Category	Feature	Description
	Syslog Configuration	Configures syslog to send logs and also configures the service- and system-related alert thresholds.

How it works

WAF performs protocol parsing and in-depth decoding on the web access traffic. It then calls the access control, rule detection, and semantic analysis engines to analyze the traffic and determines whether to allow or block the traffic based on the preset policies. Besides, WAF provides a good human-machine interaction interface for administrators to adjust protected websites and security policies.

Scenarios

WAF can be used for web application protection in fields such as government, finance, insurance, e-commerce, online to offline (O2O), Internet Plus, and games. It provides the following features:

- Prevents website data leaks caused by SQL injections.
- Mitigates HTTP flood attacks by blocking a large number of malicious requests. This ensures the availability of your website.
- Prevents website defacement arising from trojans to ensure the credibility of your website.
- Provides virtual patches that enable quick fixes for newly discovered vulnerabilities.

16.4.1.7. Security Operations Center (SOC)

Security Operations Center (SOC) provides security administrators with centralized management of all users and the platform and analysis functions of Apsara Stack logs.

Features

SOC provides the following features:

Feature	Description
Dashboard	Allows you to view the overall security statistics and perform operations.
Security monitoring	Allows you to view the security events of all users and the platform.
Asset management	Allows you to view the security status of user assets and platform assets.
Log analysis	Analyzes logs from multiple data sources, detects unexpected alerts, and improves alert detection of Apsara Stack.
Report management	Allows you to quickly export reports for various purposes.
System configurations	Allows you to configure system features such as alerts, updates, global policies, and account management.

Scenarios

- Scenario 1: routine monitoring

SOC regularly inspects system security. Currently, SOC focuses on security issues on the users. The following features are provided:

  - Urgent risk detection: checks for urgent security risks on a daily basis. Security risks include user security alerts, vulnerabilities, and server configuration risks.
  - Risk management: identifies and handles high-risk security alerts, vulnerabilities, and server configuration risks.
  - Attack data collection: shows the number of attacks and attack protection information.
  - Security reports: sends daily, weekly, or monthly security reports to users.

- Scenario 2: security evaluation for new assets

  Monitors asset changes, detects new assets, and evaluates asset security. Generates security evaluation reports on new assets to help you determine whether to add these assets to your network. The following features are provided:

  - Scans vulnerabilities on servers and web applications.
  - Verifies server configurations.
  - Performs baseline check on cloud services.

- Scenario 3: urgent event handling and cause tracking

  After an urgent event is detected, SOC handles the event and tracks the event cause.

16.4.1.8. On-premises security operations services

To ensure the stability, reliability, security, and regulatory compliance of the cloud platform, Apsara Stack Security Standard Edition provides multiple security products and on-premises security operations services to ensure the availability, confidentiality, and integrity of the systems and data of users. Security operations services are indispensable in the security system. The combination of security products and security operations services gives full play to the security features of both Apsara Stack products and Apsara Stack Security products, and enhances the security of the Apsara Stack network environment from both technology and management aspects.

On-premises security operations services aim to help users use the security features of both Apsara Stack products and Apsara Stack Security products to protect the user applications. Security operations services include services that cover the entire security lifecycle of Apsara Stack user businesses, such as pre-release security assessment, access control policy optimization, periodic security assessment, routine security inspection, and emergency response. These services help users create a cloud security operations system to enhance the application system security and ensure secure and stable businesses.

Services

On-premises security operations services are as follows:

On-premises security operations services

Category	Service	Description

Category	Service	Description
User business security operations	User asset research	With the authorization of a user, this service periodically researches the cloud businesses of the user and develops a business list containing information such as the business system name, ECS, RDS, IP address, domain name, and owner.
	New business security assessment	• Before a user migrates a new business system to the cloud, this service detects system vulnerabilities and application vulnerabilities in the new business system using both automation tools and manual operations. • Provides advice and verification on vulnerability fixes.
	Periodic business security assessment	• Periodically uses automation tools to detect system vulnerabilities, application vulnerabilities, and security risks in running businesses. • Provides advice on handling detected risks, including but not limited to security policy settings, patch updates, and application vulnerability handling.
	Access control management	Provides inspection and guidance on applying access control policies when a new business is migrated to the cloud.
	Access control routine inspection	Periodically checks for access control risks of user businesses.
	Security risk routine inspection	Monitors and inspects security events in Apsara Stack Security. Informs the user of verified events and provides advice on event handling.
Apsara Stack Security operations	Rule update	Periodically updates the rule libraries of Apsara Stack Security products.
	Product integration	• Provides support for integrating Apsara Stack Security products with the application systems of users. • Helps users customize and optimize security policies.
Security event response	Event alerts	Synchronizes recent security events information from Alibaba Cloud, and helps users remove the risks.
	Event handling	Handles urgent events such as attacker intrusions.

Service output

On-premises security operations services output the following documents:

- Weekly, monthly, and yearly service reports
- Asset lists
- System security check reports

SLA

The SLA terms of on-premises security operations services are as follows:

- **Asset management**: Update the asset list once a month.
- **Security event response**: Respond within 30 minutes during work hours.
- **Security check**:
  - Complete a pre-release security check within two workdays.
  - Perform a periodic security check once a quarter.

Duties

Partners authorized by Alibaba Cloud provide on-premises security operations services, and Alibaba Cloud provides service quality management and technical support.

Owner	Duties
Alibaba Cloud	• Assign and manage tasks of service providers and on-premises engineers. • Assess the services provided by service providers and on-premises engineers. • Train service providers and on-premises engineers and provide technical support. • Provide project coordination and process and quality management.
Service provider	• Perform security check and routine inspection on the system of the user. • Provide advice on fixing vulnerabilities. • Maintain the access control policies of the user resources. • Update and maintain the security rules and policies of Apsara Stack Security. • Respond to security events. • Provide security technical support for users.
User	• Authorize service providers to perform security operations. • Follow the security advice to carry out the security plans on businesses. • Improve the security system.

Risk control

The following measures are taken to control risks in on-premises security operations services:

Category	Risk Item	Measure
Engineer and	Organization	Only Alibaba Cloud and authorized enterprises can provide security services.

organization qualification Category	Risk Item	Measure
	Engineers	All engineers must be assessed and trained by the Alibaba Cloud security team.
Confidentiality	Confidentiality agreements	All enterprise and individual service providers must sign a confidentiality agreement.
Service tool security	Tool selection	Only security tools specified by Alibaba Cloud are allowed.
	Tool use	Apply standard configurations to avoid risks in using the tools.
Operation security	Operation procedure	Perform at-risk operations, such as scanning, in batches.
	Risk notification	Inform the users of risks in the operations, and provide risk avoidance and control methods. Perform operations only with the consent of the users.

16.4.2. Optional security services

In addition to the security services provided by Apsara Stack Security Standard Edition, multiple optional security services are also provided to meet various security needs. We recommend that you choose optional security services based on your business needs.

16.4.2.1. DDoS Traffic Scrubbing

Backed by its large-scale and distributed operating system and more than a decade of experience in defending against security attacks, Alibaba Cloud has designed and developed the DDoS Traffic Scrubbing module based on the cloud computing architecture to protect the Apsara Stack platform against large amounts of distributed denial of service (DDoS) attacks.

Features

The following table describes the features provided by the DDoS Traffic Scrubbing module.

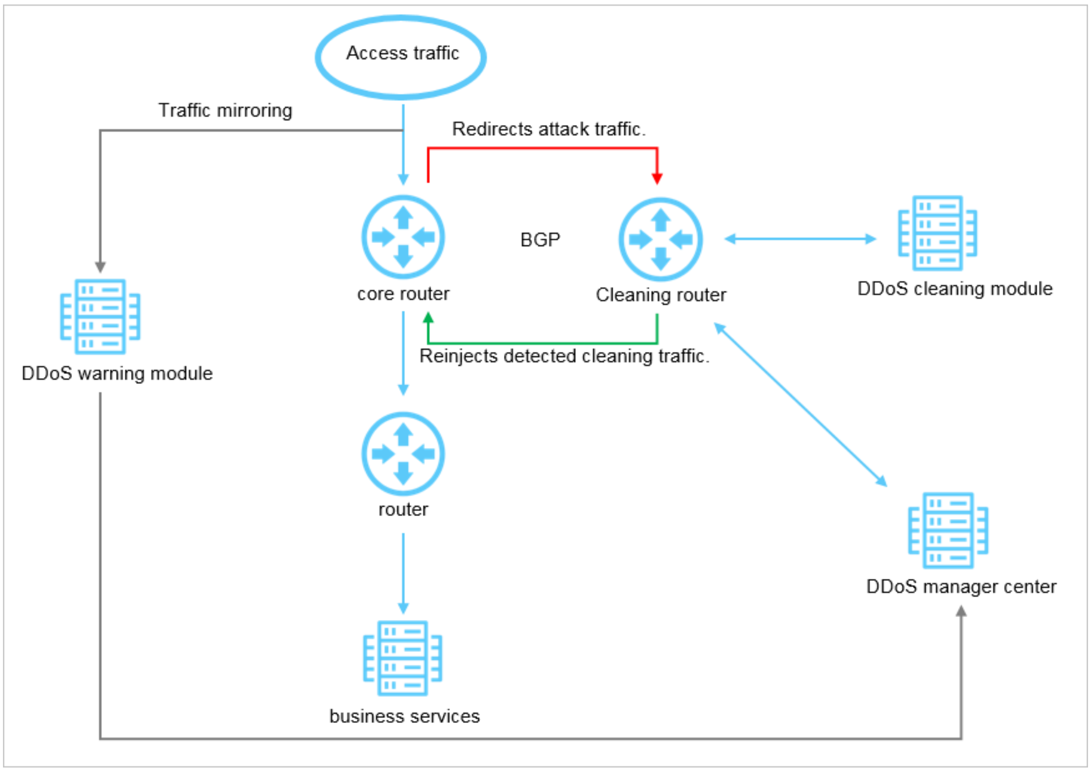
Feature	Description
Traffic scrubbing against DDoS attacks	Detects and prevents attacks such as SYN flood, ACK flood, ICMP flood, UDP flood, NTP flood, DNS flood, and HTTP flood.
DDoS attack display	Allows you to view DDoS attacks in the console and search for DDoS attacks by IP address, status, and event information.
DDoS traffic analysis	Allows you to monitor and analyze the traffic of a DDoS attack, and view the attack traffic protocol and the top 10 IP addresses that have launched most attacks.

How it works

After the Traffic Security Monitoring module detects unusual traffic, the DDoS Traffic Scrubbing module reroutes, scrubs, and reinjects the traffic, as shown in Traffic scrubbing. This mitigates DDoS attacks and ensures normal running of businesses.

Traffic scrubbing

Traffic scrubbing



The Traffic Security Monitoring module sends information about the detected DDoS attacks to the DDoS Traffic Scrubbing module. The DDoS Traffic Scrubbing module is connected to the border gateway device. When a DDoS attack is detected, this module configures a Border Gateway Protocol (BGP) path for the border gateway to reroute the attack traffic to the DDoS Traffic Scrubbing module. The DDoS Traffic Scrubbing module then scrubs the traffic based on the configured scrubbing policies, filters out unusual traffic, and reinjects the normal traffic to the border gateway.

Note Apsara Stack Security cannot scrub the traffic between internal networks.

Advantages

The DDoS Traffic Scrubbing module has the following feature advantages:

- **Detection of all common DDoS attacks**

  This module protects you from various DDoS attacks, such as HTTP flood, SYN flood, UDP flood, UDP DNS query flood, stream flood, ICMP flood, and HTTP GET flood, at the network layer, transport layer, and application layer. This module also informs you of the website defense status through real-time SMS messages.

- **Automatic response to attacks within one second**

  This module uses the world leading attack detection and prevention technologies. It can complete the protection process within one second, covering attack discovery, traffic rerouting, and traffic scrubbing. This module triggers traffic scrubbing when the traffic scrubbing thresholds are violated or when DDoS attacks are detected during network behavior analysis. This reduces network jitter and ensures the availability of your businesses in the case of DDoS attacks.

- **High scalability and high redundancy of anti-DDoS capabilities**

With high scalability and high redundancy of the cloud computing architecture, this module can be easily scaled up to realize high scalability of anti-DDoS capabilities.

- **Bidirectional protection to avoid the abuse of cloud resources**

  This module not only protects your system against external DDoS attacks but also detects resource abuse in your cloud environment. If any of your cloud resources in Apsara Stack is used to launch DDoS attacks, the Traffic Security Monitoring module will cooperate with Server Guard to restrict the network access of the hijacked resource and generate an alert.

16.4.2.2. Cloud Firewall

Cloud Firewall manages north-south traffic in a centralized manner and provides access control and traffic analysis features to better protect your network.

Features

The following table describes the features provided by Cloud Firewall.

Category	Feature	Description
Internet Firewall	Access Control	Supports Internet firewalls. You can configure outbound and inbound policies, including the access source, destination type, destination, protocol, port type, port, application, and policy action.
	Firewall Switch Policy	Allows you to search for target assets by asset type, region, instance ID, and IP address. You can enable firewall policies, including Internet Firewall, VPC-VPC, and IDC-VPC policies.
	Intrusion Prevention Policies	Allows you to set the threat engine mode to the monitoring mode or traffic control mode, to configure an IP address whitelist for outbound and inbound policies, and to customize basic policies for basic protection. You can use the Virtual Patches function and turn on the function at one click. This feature protects your network against abnormal connections, command execution, brute-force cracking, scanning, information leakage, distributed denial of service (DDoS) attacks, overflow attacks, web attacks, backdoors, trojans, worms, mining, and reverse shells.
	Event Log	Allows you to search for event logs by source IP address, destination IP address, event type, action, and time range.
	Traffic Log	Allows you to search for traffic logs by different conditions.
	VPC Firewall	Detects and controls communication traffic between two VPCs. You can configure VPC firewall policies, including the access source, destination type, protocol type, port type, application, and policy action.
	Internal Firewall	Controls inbound and outbound traffic between ECS instances. You can configure internal firewall policies, including the access source, destination, protocol type, port range, and action.

Category	Feature	Description
VPC Firewall	IDC-VPC Firewall	You can configure IDC-VPC firewall policies, including the access source, destination type, protocol type, port type, application, and action.
	Firewall Switch Policy	Allows you to search for target assets by asset type, region, instance ID, and IP address. You can enable the firewall policies, including Internet Firewall, VPC-VPC, and IDC-VPC policies.
	Intrusion Prevention Policies	Allows you to set the threat engine mode to the monitoring mode or traffic control mode, to configure an IP address whitelist for outbound and inbound policies, and to customize basic policies for basic protection. You can use the Virtual Patches function and turn on the function at one click. This feature protects your network against abnormal connections, command execution, brute force cracking, scanning, information leakage, distributed denial of service (DDoS) attacks, overflow attacks, web attacks, backdoors, trojans, worms, mining, and reverse shells.
	Event Log	Allows you to search for event logs by source IP address, destination IP address, event type, action, and time range.
	Traffic Log	Allows you to search for traffic logs by different conditions.

Scenarios

Cloud Firewall is applicable to the following scenarios:

- Control the access traffic from the Internet to ECS instances: For example, a financial company on Alibaba Cloud uses IPS to protect their HTTP and other businesses exposed on the Internet.
- Prevent command-and-control activities: For example, a governmental organization on Alibaba Cloud analyzes not only the access traffic from the Internet to ECS instances but also the command-and-control traffic from ECS instances to the Internet. Based on the analysis, the organization can determine which ECS instances are at risk and block anomalous access in real time to avoid potential risks.

Benefits

- Firewall as a service (FWaaS), which is easy to use

  Cloud Firewall uses the SDN technology. You can use Cloud Firewall after a simple policy configuration. Cloud Firewall helps you get rid of the basic but complex system and network configurations such as image installation and routing setup that are required by traditional firewalls. In addition, you do not need to be concerned about issues such as disaster recovery, capacity expansion, and deployment.

- Stability and reliability

  Cloud Firewall is deployed in dual available zone (AZ) mode. The failure of any server or AZ does not cause Cloud Firewall to break down.

- Centralized policy management

Cloud Firewall provides complete north-south traffic control for your assets. You can fully control access to your ECS instances and isolate ECS instances for security.

With Cloud Firewall, you can control access to common cloud assets such as ECS, RDS, and SLB instances at the network level and resolve anomalous access issues.

- Real-time intrusion prevention

  With the built-in IPS, Cloud Firewall can receive simultaneous updates of network-wide threat intelligence, and detect and block threats from the Internet in real time.

- Business relationship visibility

  Cloud Firewall shows assets and their access relationships in topology views. After you subscribe to the Cloud Firewall service, you can gain instant visibility of your business regions, groups, assets, access relationships between assets, and clustering analysis of user traffic without any configurations. Cloud Firewall supports visual analysis of traffic to maximize the correctness of policies.

16.4.2.3. SDDP

Sensitive Data Discovery and Protection (SDDP) is a data security service that detects and protects sensitive data in Apsara Stack big data services.

SDDP uses the big data analytics capabilities and AI technologies of Alibaba Cloud to detect and classify sensitive data based on your business requirements. SDDP can also mask sensitive data both in transit and at rest, monitor dataflows, and detect abnormal activities. SDDP uses precise detection and analysis to provide visible, controllable, and industry-compliant protection for your sensitive data. SDDP can detect and protect sensitive data in a variety of Apsara Stack services, such as MaxCompute, Object Storage Service (OSS), AnalyticDB for MySQL, Tablestore, and ApsaraDB RDS.

Features

The following table describes the features of SDDP.

Feature		Description
Classification and detection of sensitive data	Detection of new data	A department administrator can authorize SDDP to scan and protect data assets based on business requirements. SDDP scans and monitors only data assets on which it has permissions.
	Classification of sensitive data	SDDP can classify sensitive data in Apsara Stack services such as MaxCompute, OSS, AnalyticDB for MySQL, Tablestore, and ApsaraDB RDS. You can define classification rules for sensitive data by using methods such as keywords and regular expressions.
	Detection of sensitive data	SDDP has built-in algorithms that detect sensitive data. SDDP uses file clustering, deep neural networks, and machine learning to detect sensitive images, text, and fields.

Feature		Description
Management of sensitive data permissions	Detection of asset permissions	SDDP can redirect you to pages that show the permissions of data assets. SDDP also allows you to view the accounts that have permissions to access the data assets. The data assets include MaxCompute projects, MaxCompute tables, MaxCompute columns, MaxCompute packages, AnalyticDB databases, AnalyticDB tables, OSS buckets, Tablestore instances, and Tablestore tables.
	Detection of account permissions	SDDP allows you to view all accounts in a department and perform a fuzzy search for departments or accounts. SDDP shows the relationships between departments and accounts in a hierarchical display.
	Detection of abnormal permission usage	SDDP automatically detects abnormal permission usage in Apsara Stack services, such as MaxCompute, OSS, AnalyticDB for MySQL, and Tablestore.
Monitoring of dataflows and operations	Dataflow monitoring	SDDP monitors dataflows among entities, including data storage services such as MaxCompute, OSS, AnalyticDB for MySQL, and Tablestore, data transmission services such as DataHub and Data Integration, the data stream processing service Blink, external databases, and external files. SDDP displays dataflows and abnormal activities on dynamic graphs. This way, you can click an abnormal activity on a graph to redirect to the page for handling the abnormal activity.
	Detection of abnormal operations on data	SDDP detects abnormal operations in Apsara Stack services, such as MaxCompute, OSS, AnalyticDB for MySQL, and Tablestore.
	Detection of abnormal dataflows	SDDP detects abnormal dataflows, such as abnormal downloads, in Apsara Stack services. The Apsara Stack services include MaxCompute, OSS, AnalyticDB for MySQL, and Tablestore.
	Custom detection rules	SDDP allows you to customize rules that are used to detect abnormal dataflows and operations based on algorithms.
Abnormal activity processing	Configuration for abnormal activity detection	SDDP allows you to configure the thresholds and rules that are used to detect abnormal activities, such as abnormal dataflows, abnormal permission usage, and abnormal data operations.
	Abnormal activity processing	SDDP provides a built-in console that you can use to process abnormal activities. In the console, you can search for abnormal activities by department, event type, account, processing status, or time of occurrence.
	Abnormal activity statistics	SDDP collects statistics on the processing of abnormal activities and then displays the statistics on a dynamic graph. The abnormal activities include abnormal dataflows, abnormal permission usage, and abnormal data operations.

Feature		Description
Static data masking	Static data masking	SDDP statically masks sensitive data in Apsara Stack services, such as MaxCompute, OSS, AnalyticDB for MySQL, Tablestore, and ApsaraDB RDS. SDDP supports the following masking algorithms: hashing, redaction, substitution, rounding, encryption, and shuffling.
Intelligent audit	Intelligent audit	SDDP collects and audits the operation logs of Apsara Stack services such as MaxCompute, OSS, and ApsaraDB RDS.

Scenarios

- Complies with laws and regulations on personal information protection

  SDDP detects personal information in large amounts of data, automatically marks risk levels for personal information, and detects data leaks. Enterprises can use SDDP to ensure that their systems comply with laws and regulations on personal information protection.

- Classifies and protects sensitive data of enterprises

  SDDP classifies and detects sensitive data, manages data permissions, and identifies abnormal activities, such as abnormal dataflows, abnormal permission usage, and abnormal data operations, based on specified rules. This allows enterprises to protect a diverse classification of sensitive data.

- Handles data leaks

  SDDP detects abnormal activities based on specified rules and allows you to aggregate and handle abnormal activities in a centralized manner. This helps enterprises handle data leaks online and allows efficient and secure O&M.

Benefits

SDDP is a data security module of Apsara Stack Security. SDDP can detect and protect sensitive data in real-time computing services such as Blink, DataHub, AnalyticDB for MySQL, and Tablestore. SDDP can also detect and protect sensitive data in offline computing services such as MaxCompute and OSS. SDDP detects the following types of sensitive data based on the same standard: structured, semi-structured, and unstructured. SDDP has the following benefits:

- Precise detection

  SDDP uses a built-in rules engine, a natural language processing model, and a neural network model based on the AI technologies and expert teams of Alibaba Cloud to precisely detect sensitive personal information, sensitive system configurations, and confidential documents from large amounts of data.

- Closed-loop management

  SDDP implements closed-loop management, which includes detection, protection, and handling to help enterprises avoid risks.

- Intelligent detection

  SDDP provides an intelligent and multi-level filtering model based on data analytics and detection capabilities of Alibaba Cloud to effectively detect abnormal activities and meet operational requirements.

- Flexible definition

SDDP allows you to customize configurations based on your business requirements. For example, you can customize the rules for sensitive data detection, the definitions of sensitive data, thresholds, and the adaptability of models that are used to detect abnormal activities.

17.Key Management Service (KMS)

17.1. What is KMS?

Key Management Service (KMS) is a one-stop service platform for key management and data encryption. KMS provides simple, reliable, secure, and standard-compliant capabilities to encrypt and protect data. KMS greatly reduces your costs of purchase, operations and maintenance (O&M), and research and development (R&D) on cryptographic infrastructure and data encryption services. This helps you focus on the business development.

KMS provides the following features:

- Encryption key hosting

  KMS supports encryption key hosting. An encryption key hosted on KMS is called a customer master key (CMK). You can manage the lifecycle of a CMK by enabling or disabling the CMK.

- BYOK

  KMS supports Bring Your Own Key (BYOK). You can import your own keys to KMS to encrypt data on the cloud. This facilitates key management. You can import the following types of keys to KMS:

  - Keys in your on-premises key management infrastructure (KMI)
  - Keys in user-managed hardware security modules (HSMs) of Data Encryption Service

  > **Note** Keys imported to managed HSMs in KMS cannot be exported by using any method because secure key exchange algorithms are used in KMS. Operators or third parties are not allowed to check the plaintext of keys.

- Automatic rotation of encryption keys

  A CMK in KMS can have multiple key versions. Each version represents an independently generated key and does not have any relation with other versions. KMS automatically rotates encryption keys. This helps you implement the best security practices and comply with audit requirements. For more information, see the Overview and Automatic key rotation topics of *Key rotation* in *User Guide*.

- Fully managed HSMs

  KMS provides fully managed HSMs. You can host keys in HSMs. Cryptographic operations are implemented in HSMs to protect key security.

  > **Note** To use this feature, you must purchase an HSM and the KMS license of the Advanced edition.

- Simple cryptographic API operations

  - KMS provides cryptographic API operations that are simpler than those for traditional cryptographic modules or cryptographic software libraries.
  - Encryption keys in KMS support authenticated encryption with associated data (AEAD) and deliver additional authenticated data (AAD) to protect data integrity. For more information, see the EncryptionContext topic of *Use symmetric keys* in *User Guide*.

- CMK aliases

KMS allows you to create CMK aliases, which facilitate CMK usage. For more information, see the Use aliases topic in *User Guide*. For example, you can use CMK aliases to manually rotate CMKs in specific scenarios.

- Resource tags

  KMS supports resource tags, which facilitate key resource management.

17.2. Features

17.2.1. Convenient key management

You can call KMS API operations or perform operations in the KMS console to manage CMKs.

- You can disable or enable CMKs at any time. After a CMK is disabled, the data encrypted by using this CMK cannot be decrypted.
- You can schedule the deletion of a CMK by specifying a waiting period. You can cancel the scheduled deletion of a CMK at any time before the waiting period ends. This prevents CMKs from being accidentally deleted.
- You can use RAM to manage permissions on CMKs and separate encryption and decryption permissions.
- You can use EncryptionContext to enhance control over keys and ciphertext.

17.2.2. Envelope encryption

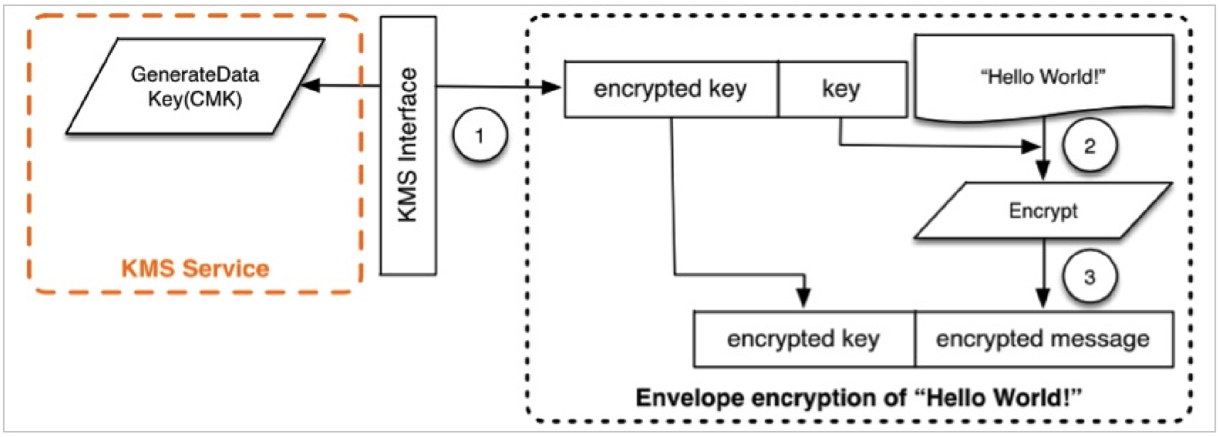
Envelope encryption is an encryption mechanism similar to the digital envelope technology. Envelope encryption allows you to encrypt data by using data keys (DKs) and encapsulate DKs in an envelope to ensure the security of their storage, transfer, and use. CMKs are not used to encrypt or decrypt data directly.

Although KMS provides the Encrypt API operation, KMS does not directly encrypt data. KMS manages CMKs and uses CMKs to encrypt and decrypt DKs. DKs are used to encrypt data.

You can use your own DK to encrypt data and then call the Encrypt API operation to encrypt the DK. You can also call the GenerateDataKey API operation to generate a DK.

Encryption process

The following figure shows the encryption process.



Encryption procedure:

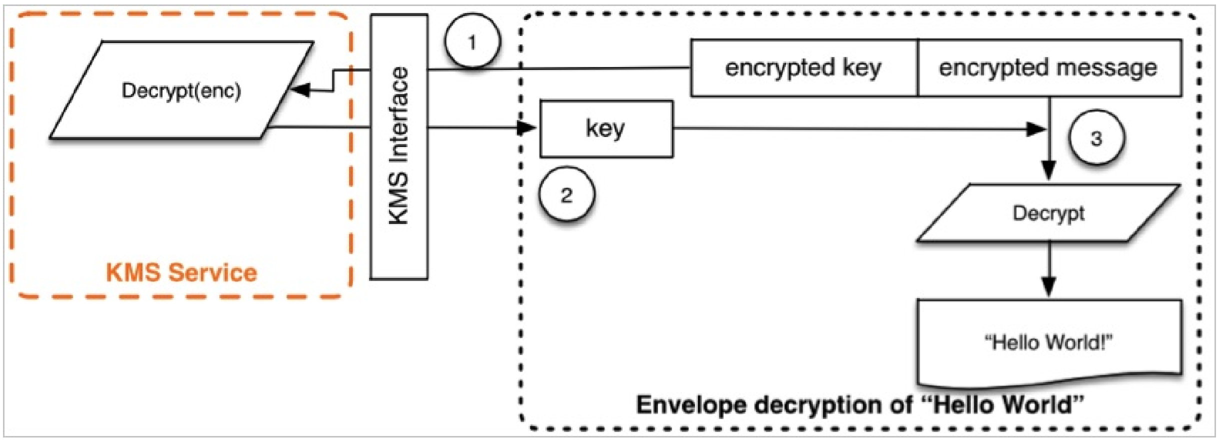
1. Use a specific CMK to generate a DK. KMS returns the plaintext and ciphertext of a DK.

   Alternatively, you can call the Encrypt operation to encrypt your own DK. KMS returns the encrypted DK.

2. Use the DK to encrypt your data. KMS returns the ciphertext of the data.
3. Store the encrypted DK and the ciphertext of the data in your storage device.

Decryption process

The following figure shows the decryption process.



Decryption procedure:

1. Use KMS to decrypt the encrypted DK. The plaintext DK is returned.
2. KMS returns the plaintext DK.
3. Use the plaintext DK to decrypt the ciphertext of your data. The plaintext data is returned.

17.2.3. Secure key storage

KMS uses the following methods to ensure key security:

- The plaintext of CMKs is stored only in the memory of hardened security appliance (HSA) modules, whereas the ciphertext of CMKs is stored only in the storage modules of KMS.
- CMKs are encrypted by using DomainKeys managed by HSA modules. DomainKeys are rotated on a daily basis.
- DomainKeys are encrypted by using a trusted computing technology and stored based on a distributed storage protocol. This ensures the high reliability of DomainKeys.

18.Apsara Stack DNS

18.1. What is Apsara Stack DNS?

Apsara Stack DNS is a service that runs on Apsara Stack to resolve domain names over internal networks, such as VPCs, data centers, and the classic network. You can configure rules to map domain names to IP addresses. Apsara Stack DNS then distributes domain name requests from clients to cloud resources, user-created business applications, business systems on your internal networks, or the business resources of Internet service providers (ISPs).

Apsara Stack DNS provides the DNS resolution and Global Server Load Balancer (GSLB) services in VPCs, data centers, and the classic network. You can perform the following operations by using Apsara Stack DNS in these internal networks:

- Access other ECS instances deployed in the same VPC.
- Access other cloud service instances on Apsara Stack.
- Access enterprise business systems.
- Access services over the Internet.
- Use the GSLB service to implement multiple-active solutions and disaster recovery, such as local active-active, local multi-active, remote active-active, active geo-redundancy, and geo-disaster recovery.
- Connect to Apsara Stack DNS with your own DNS servers over a leased line to achieve hybrid cloud integration for your business.

18.2. Benefits

As a key network service, Apsara Stack DNS controls data flows that go through Apsara Stack, resolves domain names, balances server loads, and connects Apsara Stack to data centers. Apsara Stack DNS offers multiple solutions for cloud environment deployment, zone high availability, server load balancing, and disaster recovery to support your IT operations.

Enterprise domain name management

Apsara Stack DNS provides management and resolution services for your domain names. It supports the following features:

- Performs forward and reverse DNS resolutions for domain names of cloud service instances, such as ECS instances.
- Performs forward and reverse DNS resolutions for your internal domain names.
- Allows you to add, modify, and delete DNS records of the following types: A, AAAA, CNAME, MX, PTR, TXT, SRV, NAPTR, CAA, and NS.
- Allows you to add multiple A, AAAA, or PTR records at a time. DNS servers randomly respond to all DNS queries through round robin to achieve load balancing.
- Allows you to add multiple A, AAAA, or CNAME records at a time. DNS servers respond to DNS queries based on the weight of each record type to achieve global traffic scheduling.

Flexible integration with data centers

Apsara Stack DNS can forward enterprise domain names and provide the following services for you to flexibly build your network and cascade DNS servers with user-created DNS servers:

- Global default forwarding
- Forwarding queries for specific domain names

Internet access from enterprise servers

Apsara Stack DNS supports recursive resolution for Internet domain names, which allows your servers to access the Internet.

Tenant isolation (DNS Standard Edition only)

Apsara Stack DNS allows you to manage private zones in VPCs, resolve internal domain names, and isolate DNS records based on organizations.

You can use Apsara Stack DNS to isolate data by VPC without the need to build your own DNS system. This helps reduce server and O&M costs.

GSLB

Global Server Load Balancer (GSLB) provides the following features on internal networks:

- Allows you to add multiple A, AAAA, or CNAME records at a time. DNS servers respond to DNS queries based on the weight of each record type to achieve global traffic scheduling.
- Supports scheduling line management. You can customize lines and their priorities to allow clients to access the nearest nodes and implement intelligent traffic scheduling based on geographical locations and application groups. This accelerates access to applications.
- Synchronizes configuration data for resolution among multiple clusters for which GSLB is activated. This feature is supported in multi-cloud scenarios.
- Supports address pool management to centrally manage enterprise applications by application service cluster.
- Supports custom global scheduling domains. You can centrally manage and code global scheduling instances based on your naming conventions.

Centralized management console

You can access DNS and any other cloud services on the Apsara Uni-manager Management Console with one account. This provides the following benefits:

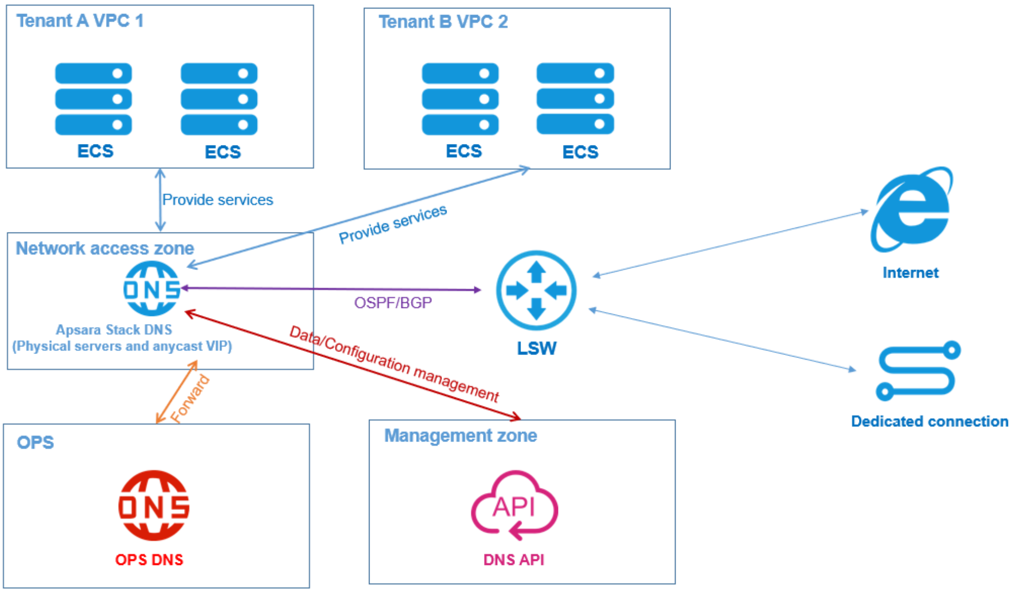
- Apsara Stack DNS supports web operations for data and service management, which facilitates your use of the DNS service.
- Apsara Stack DNS is deployed on clusters. You can add more clusters based on your needs.
- You can deploy Apsara Stack DNS in multiple zones. Apsara Stack DNS supports local active-active and zone-disaster recovery.
- Apsara Stack DNS is deployed in anycast mode, which delivers high availability and disaster recovery.

API operations

Apsara Stack DNS provides API operations so that you can integrate it with other systems.

18.3. Architecture

Architecture of Apsara Stack DNS



> **Note** Different from Apsara Stack Enterprise, Apsara Stack Agility allows you to use ZStack to create computing resources and VPCs. The architecture of Apsara Stack DNS in Apsara Stack Enterprise is different from that in Apsara Stack Agility. The following sections describe the architecture in the two editions.

Apsara Stack Enterprise

Architecture of Apsara Stack DNS (DNS Basic Edition and DNS Standard Edition)

- Uses two independent physical machines that are deployed in the network access zone to improve service availability. Apsara Stack DNS in this architecture can be scaled in or out.
- Issues anycast virtual IP address (VIP) routing requests over the LAN switch (LSW) by using Open Shortest Path First (OSPF) or Border Gateway Protocol (BGP). Anycast VIPs provide DNS services for VPCs and the classic network of tenants. The outbound IP address configured on the DNS servers can be used to forward requests to the OPS DNS server, Internet, or a dedicated enterprise network based on forwarding and recursive rules.
- Manages data and configurations by using APIs in the management zone.
- Allows you to create and query domain names on a web UI, forwards requests for cloud service domain names to the OPS DNS server, performs recursive DNS queries for Internet domain names, allows you to add, modify, delete, and query authoritative domain names and forwarding domain names of private zones, and binds and unbinds a private zone to and from a VPC.

Architecture of Apsara Stack DNS (DNS Lightweight Basic Edition)

- Supports the deployment with two physical machines on the OPS3 or OPS4 base, which eliminates the need to apply for an independent physical machine. The two physical machines achieve high availability. Apsara Stack DNS in this architecture cannot be scaled in or out.
- Issues anycast VIP routing requests over the LSW by using OSPF or BGP. Anycast VIPs provide DNS services for VPCs and the classic network of tenants. The outbound IP address configured on the DNS servers can be used to forward requests to the OPS DNS server, Internet, or a dedicated enterprise network based on forwarding and recursive rules.

- Manages data and configurations by using APIs in the management zone.
- Allows you to create and query domain names on a web UI, forwards requests for cloud service domain names to the OPS DNS server, and performs recursive DNS queries for Internet domain names.

Architecture of Apsara Stack DNS (internal GTM Standard Edition)

- Depends on the deployment of DNS Basic Edition or DNS Standard Edition. Apsara Stack DNS of the internal GTM Standard Edition is deployed on the two physical machines of DNS Basic Edition or DNS Standard Edition in the network access zone. Apsara Stack DNS in this architecture can be scaled in or out.
- Issues anycast VIP routing requests over the LSW by using OSPF or BGP. Anycast VIPs provide DNS services for VPCs and the classic network of tenants.
- Manages data and configurations by using APIs in the management zone.
- Allows you to manage domain names on a web UI, allows you to add, modify, delete, and query address pools, access policies, and scheduling instances. You can also create and delete Global Traffic Manager (GTM) synchronization clusters.

Apsara Stack Agility

Architecture of Apsara Stack DNS (DNS Basic Edition)

- Uses two independent physical machines that are deployed in the network access zone to improve service availability. Apsara Stack DNS in this architecture can be scaled in or out.
- Issues anycast VIP routing requests over the LSW by using OSPF or BGP. Anycast VIPs provide DNS services for tenants in VPCs or in the classic network. The outbound IP address configured on the DNS servers can be used to forward requests to the OPS DNS server, Internet, or a dedicated enterprise network based on forwarding and recursive rules.
- Manages data and configurations by using APIs in the management zone.
- Allows you to create and query domain names on a web UI, forwards requests for cloud service domain names to the OPS DNS server, and performs recursive DNS queries for Internet domain names.

18.4. Features

1. 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. Changes to these configurations take effect on all VPCs and the classic network.

This feature provides the same global DNS resolution service to all servers in VPCs. DNS servers use anycast IP addresses within a region. This way, seamless service failover and failback can be achieved in a specific region where data centers support disaster recovery. Note: If you do not need to upgrade Apsara Stack DNS to the Standard Edition, you can configure DNS server addresses as global anycast IP addresses to implement seamless service failover and failback over the entire network if data centers support disaster recovery.

(1) Global internal domain names

Allows you to register, search, and delete global internal domain names and add descriptions for these domain names. You can also add, delete, and modify DNS records. The following DNS record types are supported: A, AAAA, CNAME, MX, PTR, TXT, SRV, NAPTR, CAA, and NS.

Allows you to add multiple A, AAAA, or PTR records at a time. DNS servers randomly respond to all DNS queries through round robin to achieve load balancing.

Allows you to add multiple DNS records of the A, AAAA, and CNAME types on one host. DNS servers respond to DNS queries based on the weight of each record type to achieve load balancing.

(2) Global forwarding configurations

Forwards domain name requests to another DNS server for resolution.

Supports global default forwarding, which forwards requests of domain names that do not have forwarding configurations to another DNS server for resolution.

Apsara Stack DNS can forward requests with or without recursion.

- Forward All Requests (without Recursion): Only the specified DNS server is used to resolve domain names. If the resolution fails or the request times out, a message is returned to the DNS client to indicate that the query failed.
- Forward All Requests (with Recursion): The specified DNS server is preferentially used to resolve domain names. If the resolution fails, the local DNS server is used.

(3) Global recursive configurations

Supports recursive resolution for Internet domain names, which enables your servers to access the Internet.

Allows you to enable, disable, or modify the global default forwarding configurations.

2. PrivateZone (DNS Standard Edition only)

The PrivateZone feature allows you to create tenant-specific domain names in VPCs. You can bind and unbind the domain names to and from VPCs as required to isolate tenants. Changes to these configurations take effect only in the VPCs to which the domain names are bound.

This feature provides personalized DNS resolution service to servers in the VPCs to which the domain names are bound. DNS servers use anycast IP addresses within a region. This way, seamless service failover and failback can be achieved in a specific region where data centers support disaster recovery.

(1) Tenant internal domain names

Allows you to register, search, and delete tenant internal domain names and add descriptions for these domain names. You can also add, delete, and modify DNS records. The following DNS record types are supported: A, AAAA, CNAME, MX, PTR, TXT, SRV, NAPTR, CAA, and NS.

Allows you to add multiple A, AAAA, or PTR records at a time. DNS servers randomly respond to all DNS queries through round robin to achieve load balancing.

Allows you to add multiple A, AAAA, or CNAME records at a time. DNS servers respond to DNS queries based on the weight of each record type to achieve load balancing.

Allows you to bind and unbind a domain name to and from a VPC.

(2) Tenant forwarding configurations

Forwards domain name requests to another DNS server for resolution.

Supports global default forwarding, which forwards requests of domain names that do not have forwarding configurations to another DNS server for resolution.

Apsara Stack DNS can forward requests with or without recursion.

- Forward All Requests (without Recursion): Only the specified DNS server is used to resolve domain names. If the resolution fails or the request times out, a message is returned to the DNS client to indicate that the query failed.

- Forward All Requests (with Recursion): The specified DNS server is preferentially used to resolve domain names. If the resolution fails, the local DNS server is used.

Allows you to bind and unbind a domain name to and from a VPC.

3. Internal Global Traffic Manager (internal GTM Standard Edition only)

Internal Global Traffic Manager (GTM) provides multi-cloud disaster recovery for your domain names. You can connect your domain names to an internal GTM instance to manage traffic loads between Apsara Stack systems.

Internal GTM supports internal Global Server Load Balancer (GSLB). This feature intelligently allocates IP addresses for DNS queries from request sources based on configured scheduling policies. It also supports multi-cloud, hybrid deployment and configuration data synchronization between cloud networks.

(1) Scheduling instance management

Allows you to manage scheduling instances. Each scheduling instance corresponds to an application instance.

Allows you to manage address pools. Each address pool corresponds to a service cluster of an application instance.

Allows you to manage scheduling domains and set the scheduling domains to which scheduling instances belong. You can centrally manage and code global scheduling instances based on your own naming conventions.

(2) Scheduling line management

Supports scheduling line management. You can customize lines and their priorities to allow clients to access the nearest nodes and implement intelligent traffic scheduling based on geographical locations and application groups. This accelerates access to applications.

(3) Data synchronization management

Allows you to manage global data synchronization links. You can create data synchronization links, manage data synchronization configurations, and view data synchronization information of multiple internal GTM services. The information includes local system information, information of cluster nodes on which data synchronization relationship has been established, and primary and secondary relationships.

Allows you to manage the messages for changes to data synchronization links, which helps you confirm request messages for primary nodes to actively add secondary nodes.

19.Log Service

19.1. What is Log Service?

19.1.1. Overview

Log Service is a unified solution for high volumes of log data, and provides log data collection, subscription, query, and transfer functions.

- Real-time collection and consumption: Log Service collects log data in real time from multiple channels through the client, APIs, tracking.js, and libraries. Data can be immediately subscribed and read after it is written. Interfaces such as Spark Streaming, Storm, and Consumer Library can be used to process data in real time.
- LogSearch: LogSearch creates indexes for log data in real time and provides real-time and powerful storage and query engines. LogSearch allows you to retrieve logs by various dimensions such as time, keyword, and context.

Log Service can automatically scale based on processing requirements. It can scale out to handle large volumes (PBs) of data.

19.1.2. Values

Log Service helps you build solutions for large volumes of log data.

Log Service is applicable to the following scenarios: data collection, real-time computing, data warehousing and offline analysis, product operation and analysis, operations and maintenance, and management.

- Data collection and consumption
- ETL and stream processing
- Event sourcing and tracing
- Log management

19.2. Benefits

19.2.1. Features

This topic describes the features of Log Service.

Real-time log collection

Log Service supports real-time log collection. You can collect data in real time by using the following methods.

- Log collection by using Logtail: stable, reliable, and secure. This method provides high performance at low resource consumption. You can use Logtail to collect logs from servers that run Linux or Windows and from Docker containers.
- Log collection by using APIs or SDKs: flexible, convenient, and scalable. You can use an API or SDK developed in multiple programming languages to collect logs from mobile devices.
- Log collection by integrating Log Service with cloud services: convenient and efficient. You can

integrate Log Service with cloud services such as Elastic Compute Service (ECS) and then collect logs from the cloud services.

- Other log collection methods: Syslog and Logstash.

Real-time log consumption

Log Service allows you to use stream computing systems to consume data. Log Service provides consumer libraries that are developed in multiple programming languages for data consumption.

- Comprehensive features: Log Service provides all of the features of Kafka. Log Service records consumption checkpoints and scales computing resources based on the volume of data reads. Log Service also allows you to specify a time range to consume data.
- Stability and reliability: After data is written to Log Service, it can be consumed in real time. Data is stored in duplicates in Log Service. Computing resources are scalable based on the volume of data reads.
- Easy to use: You can use Spark Streaming, Storm and SDKs to consume data. Log Service provides consumer libraries that allow you to consume data in load balancing mode.

Log query

Log Service allows you to create indexes for log data and query the data in real time. You can specify a time range and query log data by keyword.

- Large scale: Log Service supports real-time indexing for petabytes of data.
- Flexible queries: Log Service supports keyword-based queries, fuzzy matching, cross-topic queries, and contextual queries.

19.2.2. Benefits

This topic describes the benefits of Log Service.

Fully managed service

- Log Service is easy to access and use.
- LogHub provides all the features of Kafka. You can store and query functional data such as monitoring and alerting data in LogHub. The data that you store or query per day can amount to petabytes.
- LogSearch/Analytics allows you to query log data, view log data on dashboards, and configure alerts.
- Log Service allows you to import data from more than 30 data sources such as the open-source software applications Storm and Spark Streaming.

Inclusive ecosystem

- The 30-odd data sources include embedded devices, web pages, servers, and programs. LogHub can also be interconnected with consumption systems such as , Storm, and Spark Streaming.
- LogSearch/Analytics provides complete query syntax and supports connection with Grafana based on the SQL-92 and JDBC protocols.

Real-time response

- LogHub: Data can be consumed immediately after being written to Log Service. Logtail acts as an agent to collect and send data to Log Service in real time.

- LogSearch/Analytics: Data can be queried immediately after being written to Log Service. If you specify multiple query conditions, data can be returned within seconds.

Complete operations of the API and SDKs

- Log Service supports custom management and secondary development.
- All Log Service features can be implemented by using SDKs or the API. SDKs in multiple programming languages are provided. You can use an SDK to manage services and millions of devices.
- The query syntax is simple and compatible with the SQL-92 standard. User-friendly interfaces are integrated into the software environment.

19.3. Architecture

19.3.1. Components

This topic describes the components of Log Service.

Logtail

Logtail is an agent that collects logs. It has the following characteristics:

- Non-intrusive file-based log collection
  - Logtail only reads log files.
  - Log collection is not intrusive.
- High security and reliability
  - Logtail can rotate log files without data loss.
  - Data that is collected by using Logtail can be locally cached.
  - Logtail retries log collection if network exceptions occur.
- Ease of use and management
  - You can configure log collection by using Logtail on the web.
  - You can configure log collection by using Logtail in the Log Service console.
- Complete self-protection mechanism
  - Logtail monitors the CPU and memory resources that it consumes in real time.
  - Logtail allows you to set an upper limit on the resources that it consumes.

Frontend servers

Frontend servers are built on the Linux Virtual Server (LVS) and NGINX servers. They have the following features:

- Support for HTTP and REST
- Scale-out

  More frontend servers can be deployed to accommodate traffic surges.
- High throughput and low latency
  - Requests are asynchronously processed. If an exception occurs when a single request is sent, other requests are not affected.

- Log data is compressed by using the LZ4 algorithm: This reduces required network bandwidth resources and increases the processing capabilities of individual servers.

Backend servers

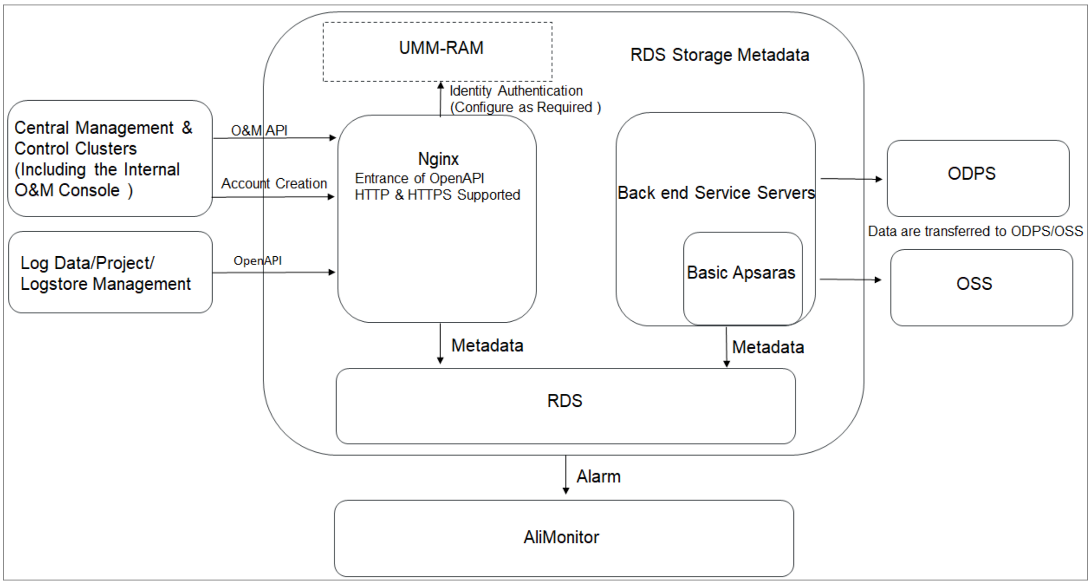
Backend processes are deployed across multiple servers. Backend servers store, index, and query data in real time. The following list describes the features of backend servers.

- High data security
    - Each log entry is stored in three copies.
    - Data is automatically recovered in the cases of disk damage or server downtime.
- Stable service
    - Logstores are automatically migrated in case of process crashes or server downtime.
    - Automatic load balancing ensures that traffic is distributed evenly among servers.
    - Strict quotas prevent some unexpected or incorrect operations of a single user from affecting other users.
- Scale-out
    - A shard is the basic unit for scale-out.
    - You can add shards to increase throughput based on your requirements.

19.3.2. System architecture

The following figure shows the architecture of Log Service.

Architecture



- The console and open APIs are located on the left side. They are used to interact with external modules.
- The core modules in the middle include:
    - UMM and RAM: account management
    - RDS: metadata storage
    - NGINX: front-end servers

- Log Service background: back-end business servers

20.API Gateway

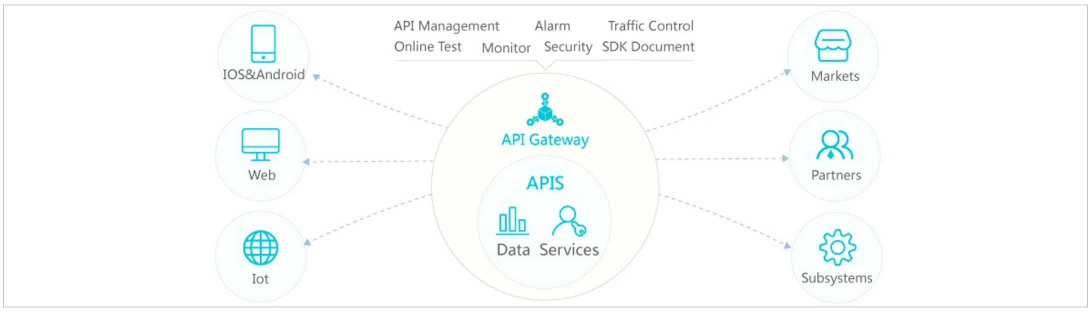
20.1. What is API Gateway?

API Gateway provides a comprehensive suite of API hosting services that help you share capabilities, services, and data with partners in the form of APIs.

- API Gateway provides multiple security mechanisms to secure APIs and reduce the risks arising from open APIs. These mechanisms include protection against replay attacks, request encryption, identity authentication, permission management, and throttling.
- API Gateway provides API lifecycle management that allows you to define, publish, and unpublish APIs. This improves API management and iteration efficiency.

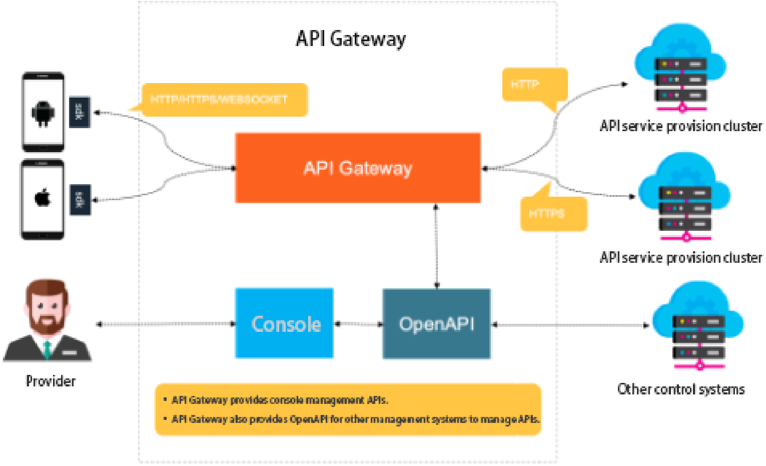
API Gateway allows enterprises to reuse and share their capabilities with each other so that they can focus on their core business.

API Gateway



20.2. System architecture

System architecture of API Gateway



- API Gateway provides console management APIs.
- API Gateway also provides OpenAPI for other management systems to manage APIs.

API Gateway consists of the following components:

- Gateway

  The gateway component is the core system that implements all business logic.

The gateway component supports multi-protocol access for all clients, including HTTP, HTTPS, and WebSocket. The gateway component manages client connections, throttles API requests, and implements IP address-based access control.

The gateway component loads user-defined APIs into the memory, processes requests from clients based on API definitions, calls back-end APIs, and returns back-end responses to clients.

- OpenAPI

  The OpenAPI component consists of a group of standard management APIs provided by API Gateway to manage API definitions. You can use the OpenAPI component to manage groups, metadata, and authorization for APIs. When the OpenAPI component receives an API change request, it synchronizes the change to all gateway services. System administrators can use the management APIs to manage the APIs that are running in the gateway in real time.

  System administrators can manage their own APIs in the API Gateway console in real time. They can also call the management APIs in their own management systems to manage their own APIs.

- Console

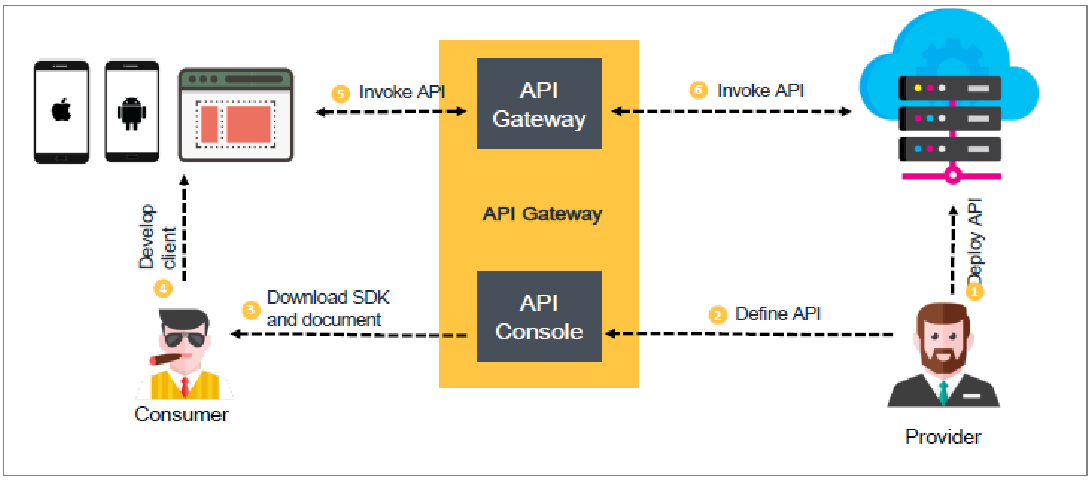
  The API Gateway console implements all features of API Gateway. System administrators can manage their own APIs in the console in real time.

  The API Gateway console provides you with a graphical user interface to call APIs through the OpenAPI component.

20.3. Features

20.3.1. API lifecycle management

API lifecycle management

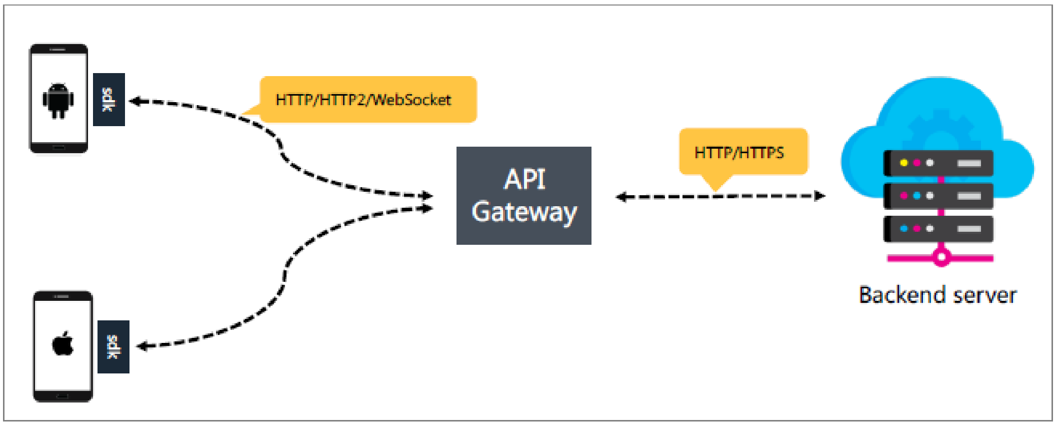


It provides a range of lifecycle management functions to publish, test, and unpublish APIs.

It provides maintenance functions such as routine API management, API version management, and quick API rollback.

20.3.2. Multi-protocol access

Multi-protocol access



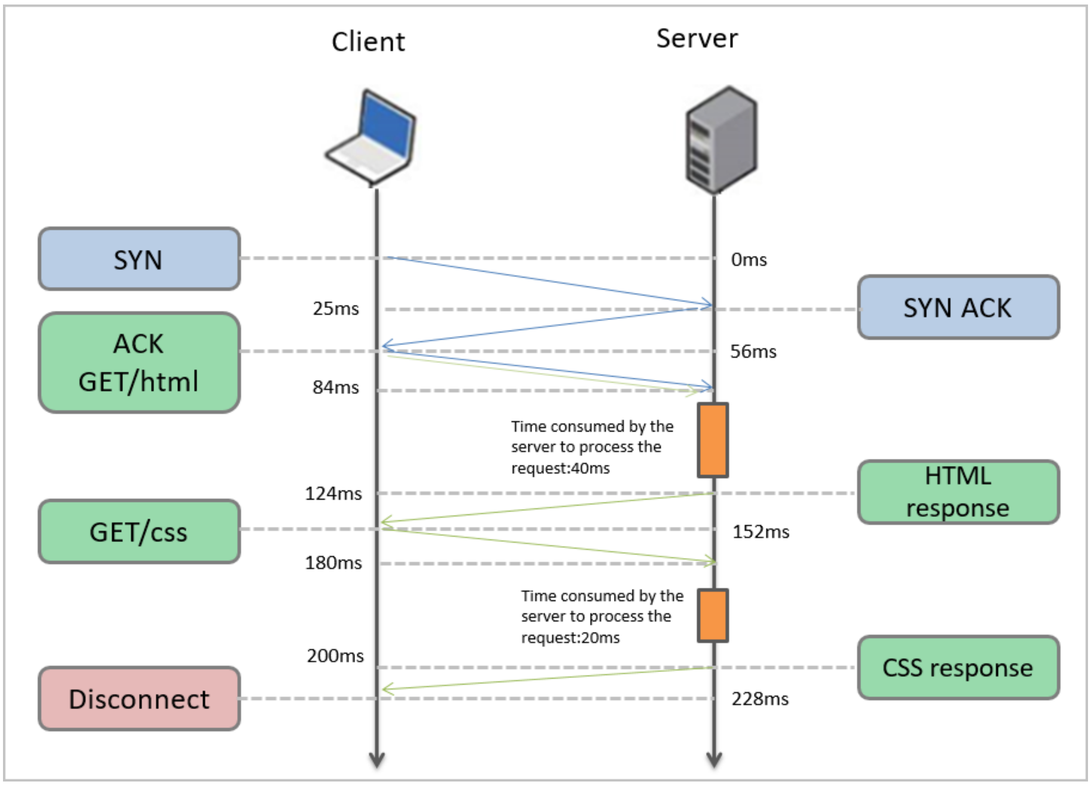
The following protocols can be selected to establish bidirectional communication between clients and API Gateway:

- HTTP: is the most popular Internet text protocol.
- HTTP/2: supports multiplexing and header compression for high efficiency.
- WebSocket: supports persistent connections for binary communications.

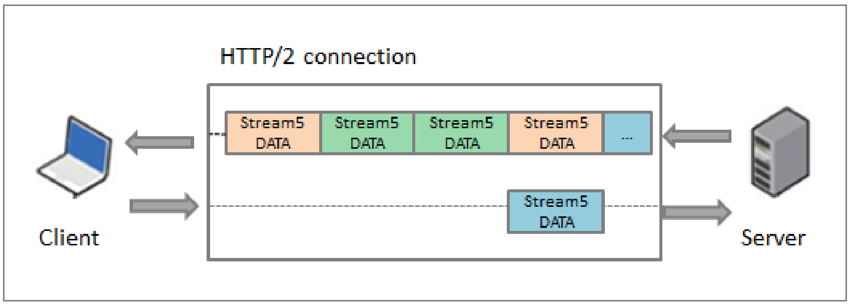
Unlike HTTP/1.x, all HTTP/2 communication is split into smaller messages and frames, each of which is encapsulated with binary encoding. In HTTP/1.x, the header information is encapsulated in the Headers frame, and the request body is encapsulated in the Data frame. A single TCP connection can be used to send multiple requests. This reduces connections to the server and improves throughput. HTTP/2 uses header compression to enable faster data transmission and deliver more benefits to the mobile network environment so that network congestion is reduced.

Browsers limit the number of HTTP/1.x connections with the same domain name at the same time. If the limit is exceeded, further requests will be blocked. The binary framing layer in HTTP/2 enables full request and response multiplexing within a shared TCP connection. An HTTP message is divided into independent frames that are interleaved and reassembled on the other end based on stream identifiers and headers. The following figures compare data transmission between HTTP/1.x and HTTP/2.

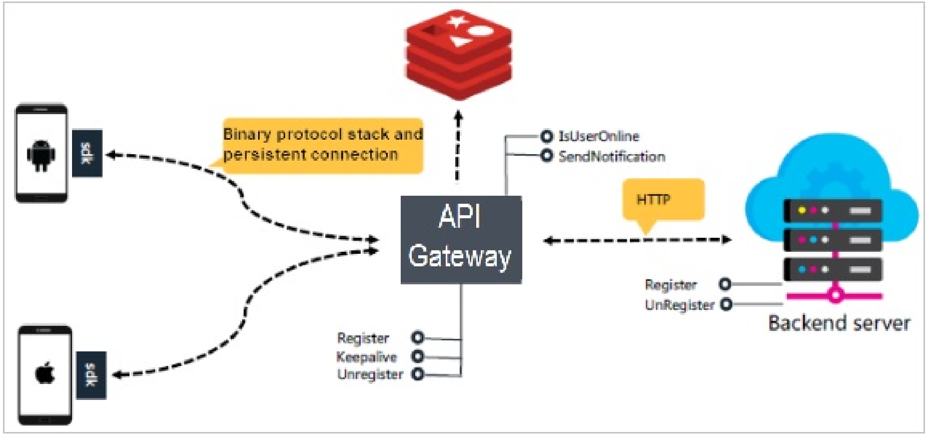
HTTP/1.x



HTTP/2



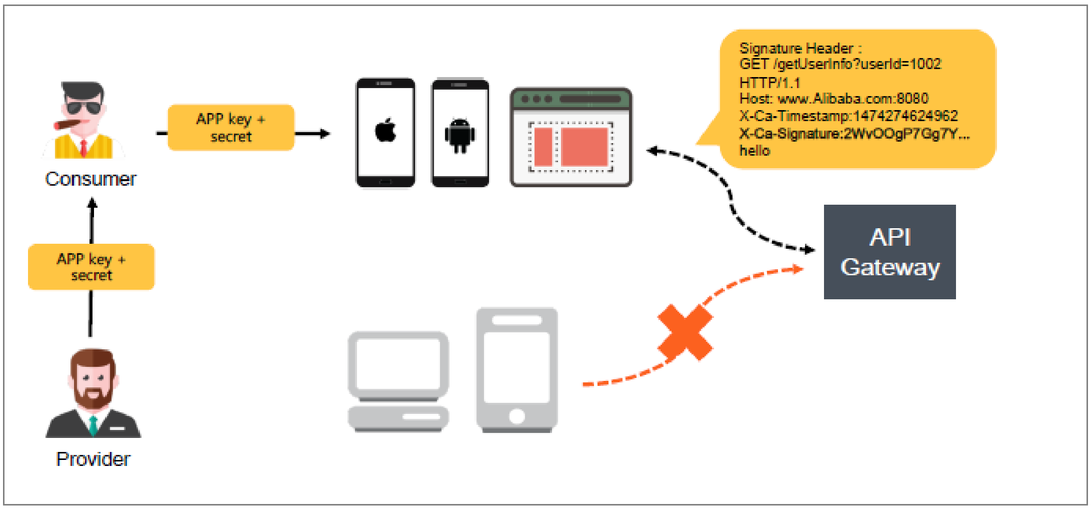
WebSocket is a protocol that enables full-duplex persistent communication. Both clients and servers can send and receive data to and from each other. HTTP is used during the handshake. After a successful handshake, clients and servers can directly communicate with each other without HTTP. The data format is lightweight, the performance overheads are low, and the communication efficiency is high. Clients can communicate with any servers without the same-origin policy.



API Gateway supports bidirectional communication and maintains persistent connections between clients and itself. API Gateway can update the online status of clients after receiving heartbeat requests. Back-end services can access the API Gateway interface to query the online status of clients and push in-application notifications. API Gateway implements bidirectional communication based on the WebSocket protocol. Bidirectional communication is supported by Android SDKs, Objective-C SDKs, and Java SDKs.

20.3.3. Application access control

Application access control


Consumer
APP key + secret
APP key + secret
Provider
Signature Header :
GET /getUserInfo?userId=1002
HTTP/1.1
Host: www.Alibaba.com:8080
X-Ca-Timestamp:1474274624962
X-Ca-Signature:2WvOOgP7Gg7Y...
hello
API Gateway

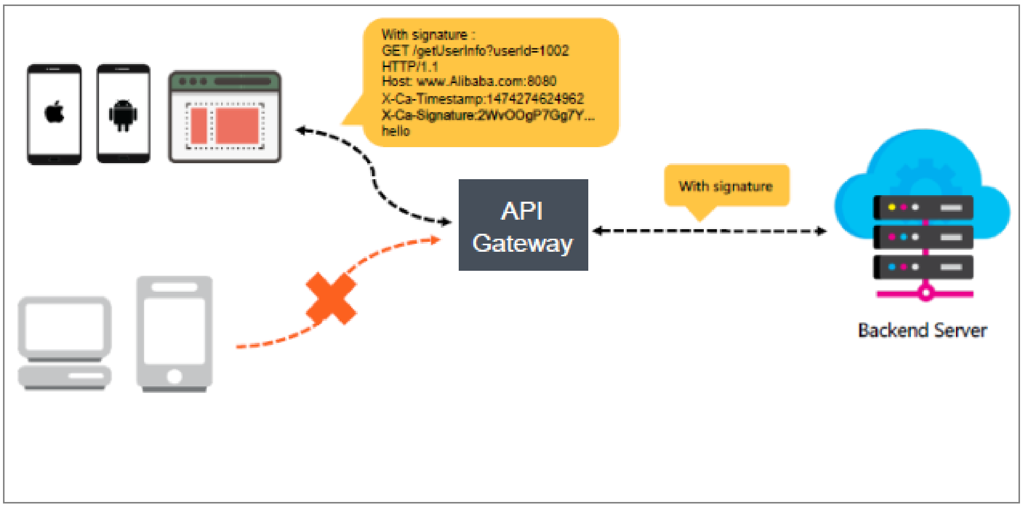
API Gateway provides an application-based authentication mechanism. This mechanism ensures that only authorized clients can send requests to the back-end services. Applications are the identities that you use to call APIs. Each application has a key pair that consists of an AppKey and an AppSecret. The AppKey parameter is added to the request header when a client sends a request, while the AppSecret is used to calculate the signature. Signature verification can protect user data from being tampered. If the verification fails, an error is reported immediately.

Symmetric encryption is used to verify the signature. The construction rules of the signature string are described in public documentation. HMAC-SHA256 is used as the signature algorithm. AppSecret is used as the encryption key that is only available to the clients and API Gateway. The client constructs and encrypts StringToSign based on specific rules. For more information about the construction methods, see Request signatures. After receiving the request, API Gateway constructs the StringToSign based on request parameters. Then, API Gateway finds the corresponding AppSecret through the AppKey. API Gateway calculates the signature string and compares it with the signature string that is sent by the user. If both signature strings are same, the signature verification is passed. Otherwise, it fails.

The principle of application-based authentication is as follows: API Gateway obtains the unique AppID based on the AppKey, and checks whether the application is authorized to access the API based on the AppID and API. If yes, access to the API is allowed. Otherwise, an unauthorized access error is reported.

20.3.4. Full-link signature verification mechanism

Full-link signature verification mechanism

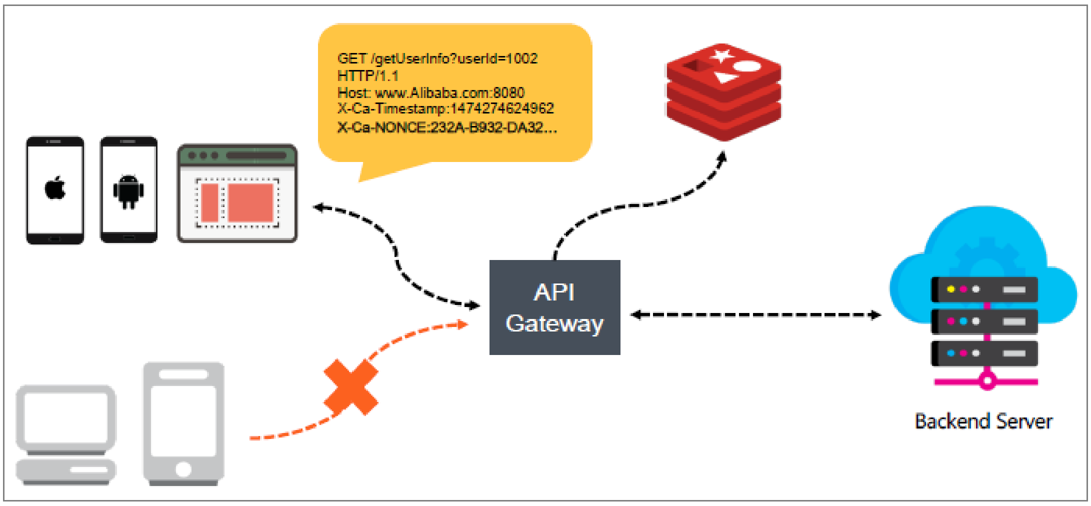


```
With signature :
GET /getUserInfo?userId=1002
HTTP/1.1
Host: www.Alibaba.com:8080
X-Ca-Timestamp:1474274624962
X-Ca-Signature:2WvOOgP7Gg7Y...
hello
```

API Gateway provides a full-link signature verification mechanism for communication between the client and API Gateway or between API Gateway and the backend service. This mechanism prevents data tampering during request transmission. When a client calls an API, the client must convert the key request data into a signature string based on API Gateway signature algorithms. The client must attach the signature string to the request header. API Gateway performs symmetric calculation to parse the signature and verify the identity of the request sender. HTTP, HTTPS, and WebSocket requests must have a signature in their header.

20.3.5. Anti-replay mechanism

Anti-replay mechanism



```
GET /getUserInfo?userId=1002
HTTP/1.1
Host: www.Alibaba.com:8080
X-Ca-Timestamp:1474274624962
X-Ca-NONCE:232A-B932-DA32...
```

API Gateway provides an anti-replay mechanism to protect against data tampering used in replay attacks.

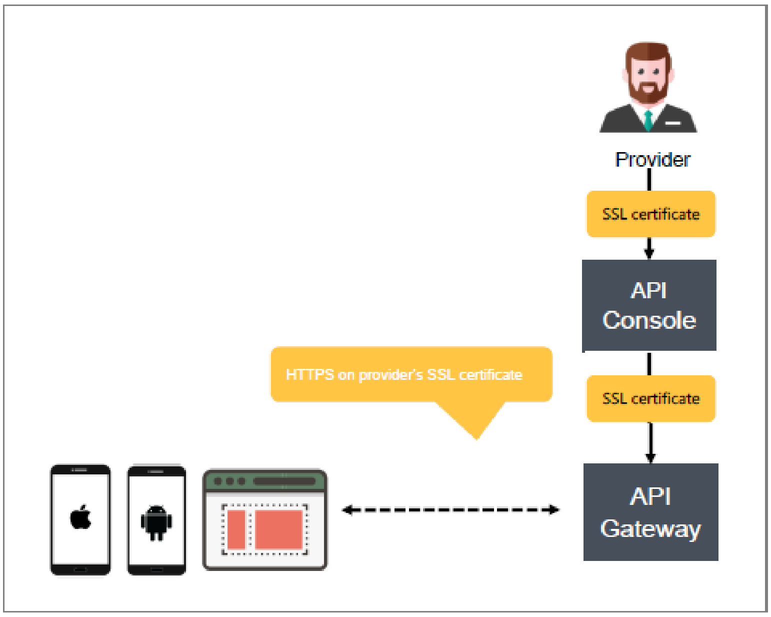
- When a client sends a request to API Gateway, the X-Ca-Nonce header is added. The value of the X-Ca-Nonce header can be any string. API Gateway verifies whether the same X-Ca-Nonce header has been passed in within 15 minutes. If yes, the request is considered a replay, and API Gateway reports an error immediately.

  A distributed cache is used. API Gateway verifies whether the same X-Ca-Nonce header exists for each request.

- The value of the Nonce parameter is included in the signature string. Therefore, it cannot be tampered.

20.3.6. HTTPS communication based on the SSL certificate of the user

HTTPS communication based on the SSL certificate of the user

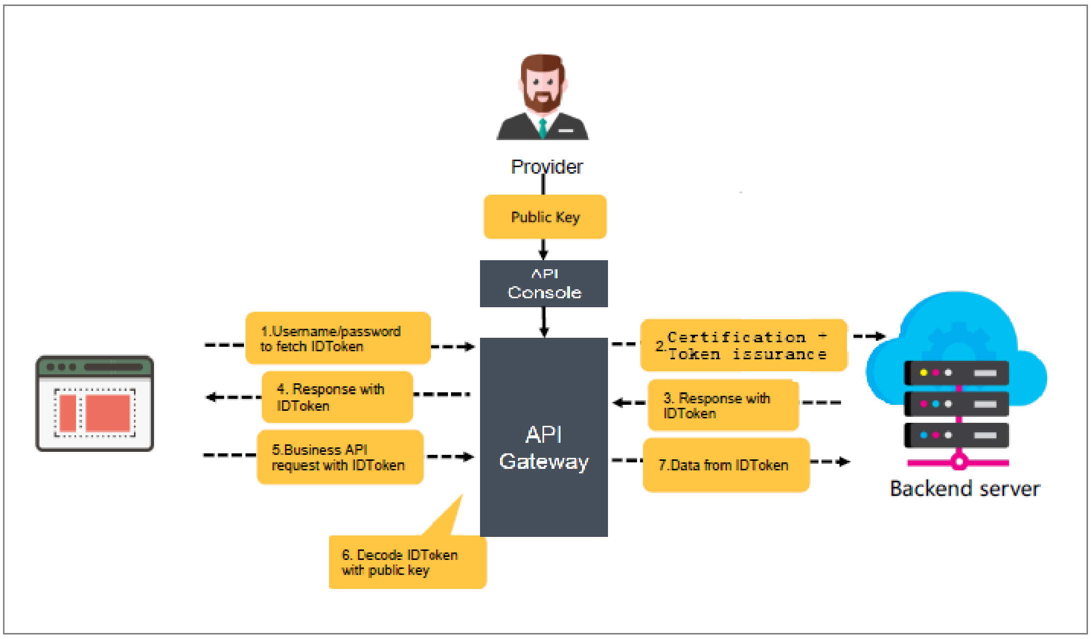


A system administrator can upload an SSL certificate corresponding to the domain name in the API Gateway console. Data transmitted between clients and API Gateway will then be encrypted based on the certificate. This prevents data tampering during transmission.

System administrators can update SSL certificates in real time in the API Gateway console.

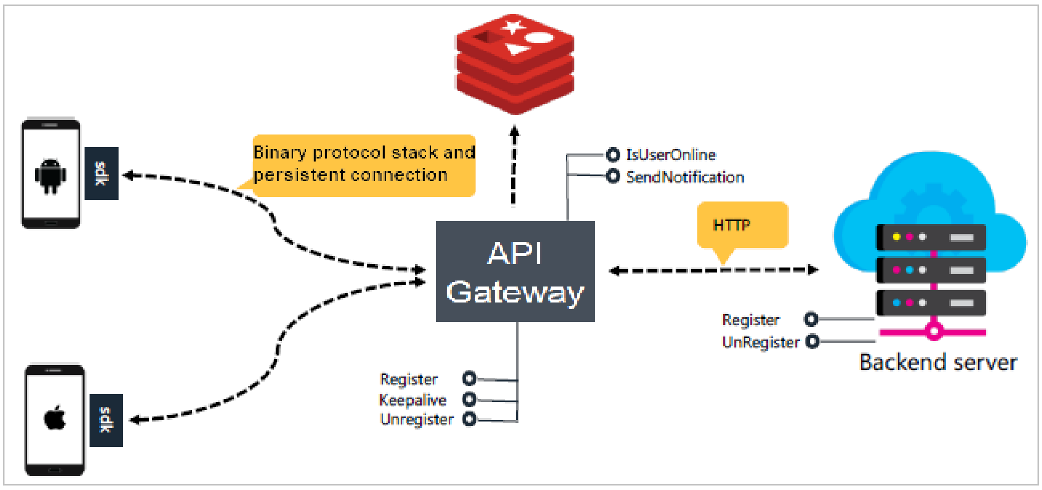
20.3.7. Support for OpenID Connect

Support for OpenID Connect


Provider
Public Key
API Console
1.Username/password to fetch IDToken
4. Response with IDToken
5.Business API request with IDToken
API Gateway
6. Decode IDToken with public key
2.Certification + Token issurance
3. Response with IDToken
7.Data from IDToken
Backend server

API Gateway supports OpenID Connect authentication, allowing API providers to verify requests based on their own user systems. OpenID Connect is a lightweight authentication standard based on OAuth 2.0. It provides a framework for identity interaction through APIs. Compared with OAuth, OpenID Connect not only authenticates a request, but also specifies the identity of the requester.

20.3.8. Bidirectional communication

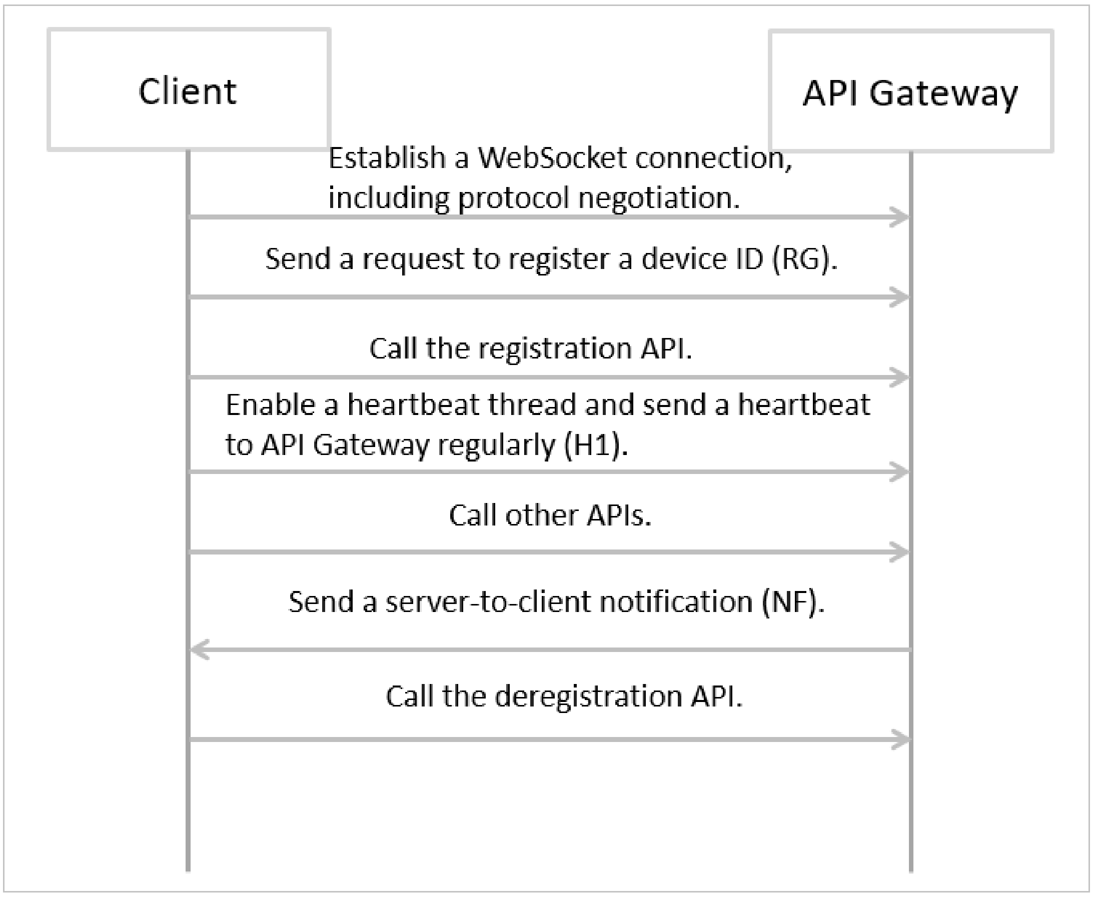


API Gateway supports bidirectional communication and maintains persistent connections between clients and itself. API Gateway can update the online status of clients after receiving heartbeat requests. Back-end services can access the API Gateway interface to query the online status of clients and push in-application notifications.

API Gateway implements bidirectional communication based on the WebSocket protocol. Bidirectional communication is supported by Android SDKs, Objective-C SDKs, and Java SDKs.

API Gateway provides built-in APIs to WebSocket users, including the APIs that clients use to register and deregister device IDs with API Gateway, and the APIs that are called to detect heartbeats.

Before establishing a WebSocket connection between clients and API Gateway, you must call the registration API to register device IDs. The following figure shows the interaction between clients and API Gateway.


Client
API Gateway
Establish a WebSocket connection, including protocol negotiation.
Send a request to register a device ID (RG).
Call the registration API.
Enable a heartbeat thread and send a heartbeat to API Gateway regularly (H1).
Call other APIs.
Send a server-to-client notification (NF).
Call the deregistration API.

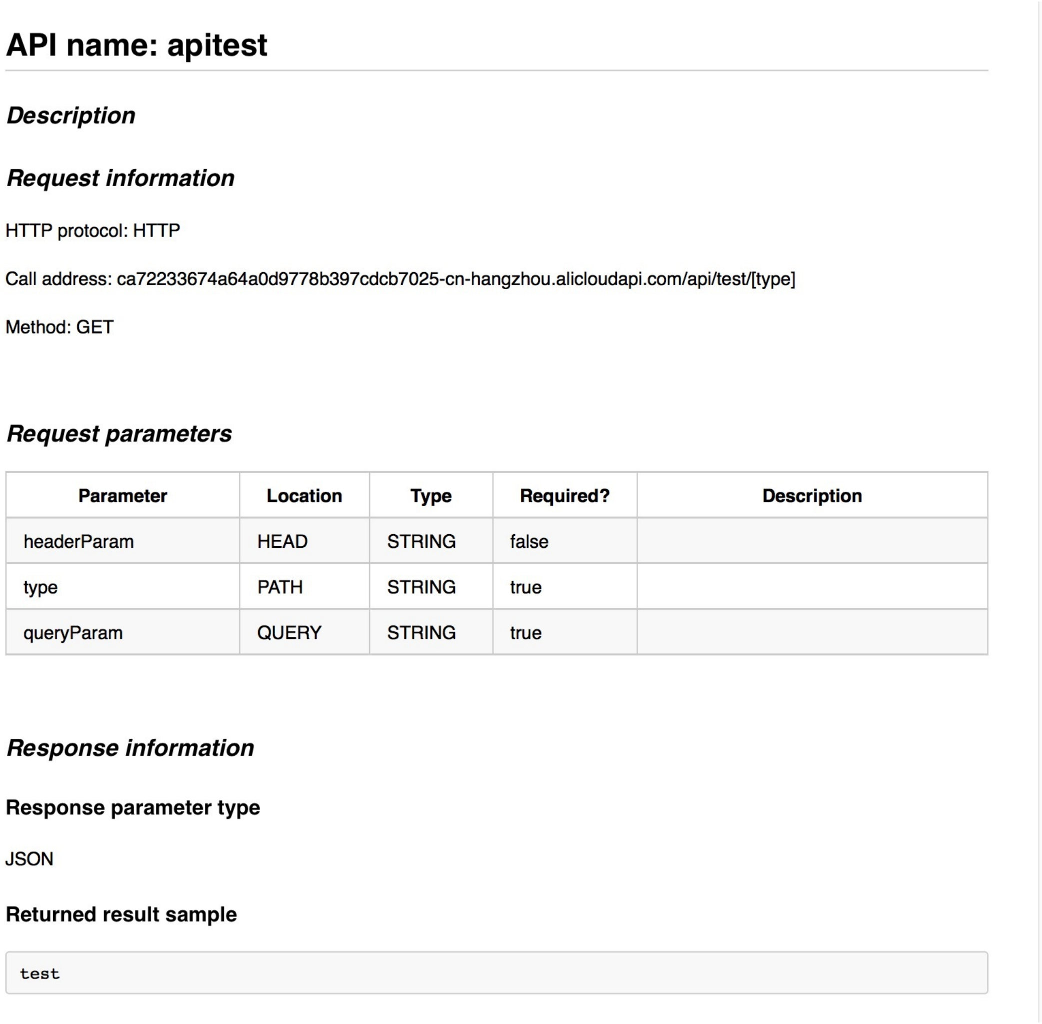
Clients need to complete the following operations:

1. Establish a WebSocket connection, including protocol negotiation.
2. Send a request to register a device ID (RG).
3. Call the registration API.
4. Enable a heartbeat thread and send a heartbeat to API Gateway regularly (H1).
5. Call other APIs.
6. Receive notifications sent by API Gateway.
7. Call the deregistration API.

20.3.9. Automatic generation of SDKs and API documentation

API Gateway can automatically generate Java, Objective-C, and Android SDKs for the APIs customized by providers. API Gateway can also generate API documentation. The following figure shows part of the API documentation.

Automatic generation of SDKs and API documentation



API name: apitest

Description

Request information

HTTP protocol: HTTP

Call address: ca72233674a64a0d9778b397cdcb7025-cn-hangzhou.alicloudapi.com/api/test/[type]

Method: GET

Request parameters

Parameter	Location	Type	Required?	Description
headerParam	HEAD	STRING	false	
type	PATH	STRING	true	
queryParam	QUERY	STRING	true	

Response information

Response parameter type

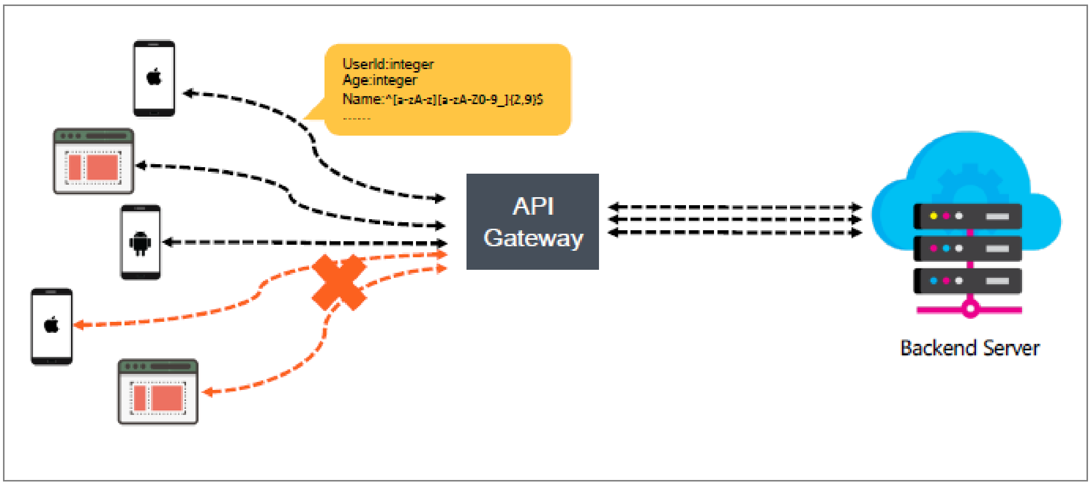
JSON

Returned result sample

```
test
```

20.3.10. Parameter cleaning

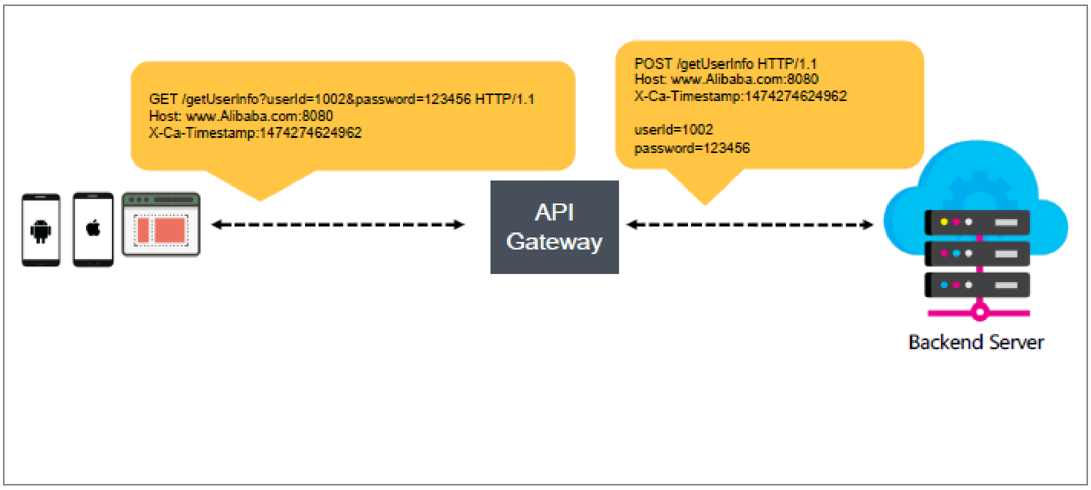
Parameter cleaning



System administrators can define the data type, regular expression, and enumeration of all API parameters. API Gateway forwards API requests that match the API definition to the backend service, while rejecting the requests that do not match the definition. This ensures that the backend service only receives standard requests that match API definitions.

20.3.11. Mappings between frontend and backend parameters

Mappings between frontend and backend parameters



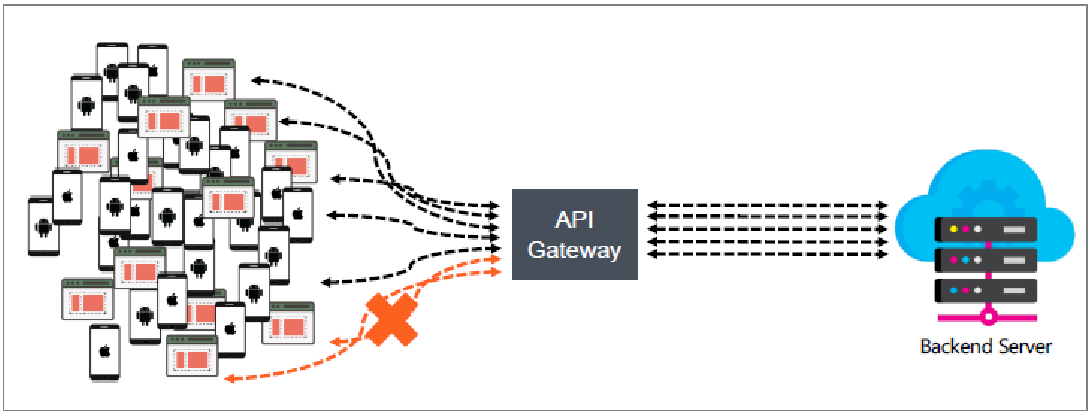
```
GET /getUserInfo?userId=1002&password=123456 HTTP/1.1
Host: www.Alibaba.com:8080
X-Ca-Timestamp:1474274624962
```

```
POST /getUserInfo HTTP/1.1
Host: www.Alibaba.com:8080
X-Ca-Timestamp:1474274624962

userId=1002
password=123456
```

API Gateway provides parameter mapping capabilities to relocate parameters within a request before sending the request to the backend service. For example, a parameter in a request sent to API Gateway is defined in Query. API Gateway can map the parameter to Form and then send the request to the backend service. This function ensures that users can access complicated backend functions by calling well-organized APIs.

20.3.12. Throttling

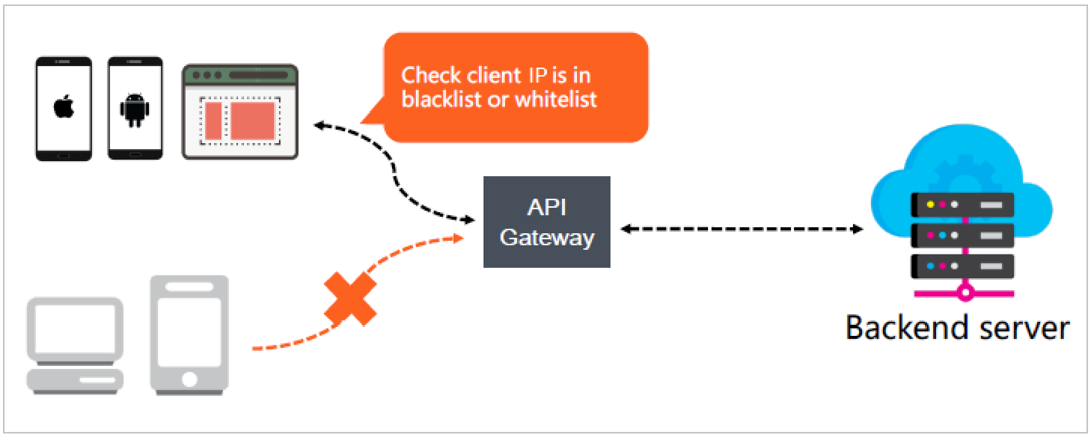


System administrators can set a request threshold based on the maximum processing capabilities of backend services. If the total number of requests exceeds the threshold, API Gateway will reject the excess requests to protect backend services from being overloaded. Supported dimensions include API, user, and application. Supported time granularity includes second, minute, hour, and day.

API Gateway uses a distributed cache, calculates the number of API requests from clients, and customizes keys in different formats based on the unit of time that you set. For example, if you set the unit of time to minute, the key is displayed in the yyyyMMddHHmm format. If the current time is 20:00 on May 7, 2019, the key is displayed as 201905072000. All requests in the current period are accumulated for this key. Requests will be recounted automatically during the next period. When another request is received, counting will be restarted.

20.3.13. IP address-based access control

IP address-based access control

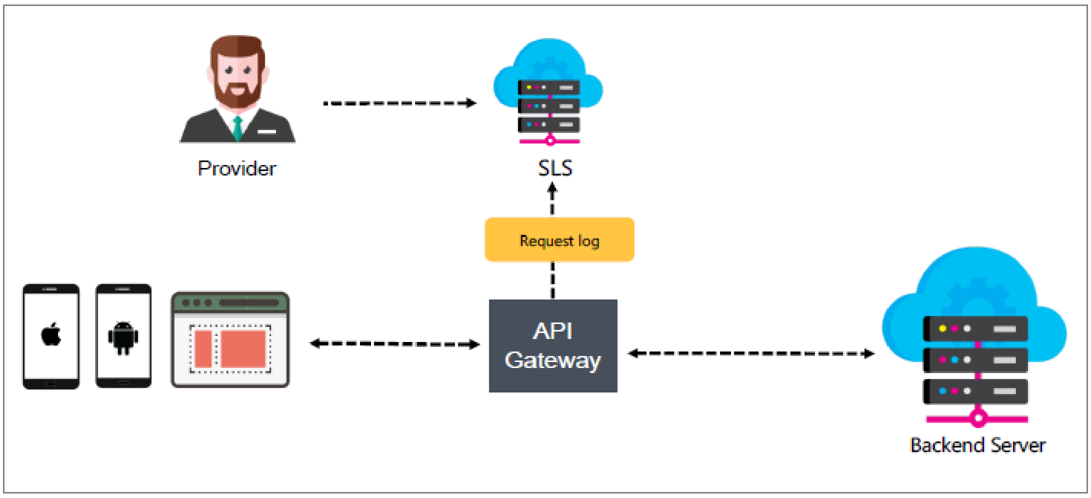


IP address-based access control is one of the API security protection measures provided by API Gateway. This measure controls the source IP addresses or IP address segments for API requests. System administrators can configure an IP whitelist or blacklist for an API to allow or deny an API request from an IP address.

The IP addresses obtained by API Gateway are egress IP addresses of clients. You cannot use the X-Forward-For header because its value can be randomly set by clients. API Gateway compares these client IP addresses with user-defined rules. It allows access to APIs from IP addresses in the whitelist and denies access to APIs from IP addresses in the blacklist.

20.3.14. Log analysis

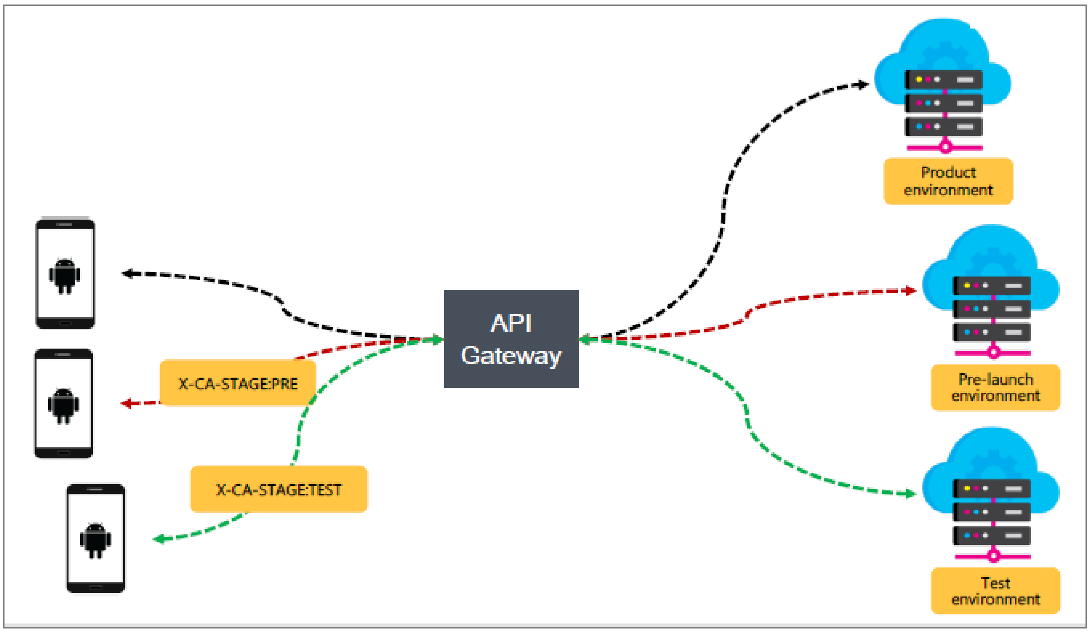
Log analysis



API Gateway sends called logs to Log Service. System administrators can use Log Service to query or download logs, or perform statistical analysis in real time. Logs can also be sent to OSS or MaxCompute.

20.3.15. Publish an API in multiple environments

Publish an API to multiple environments

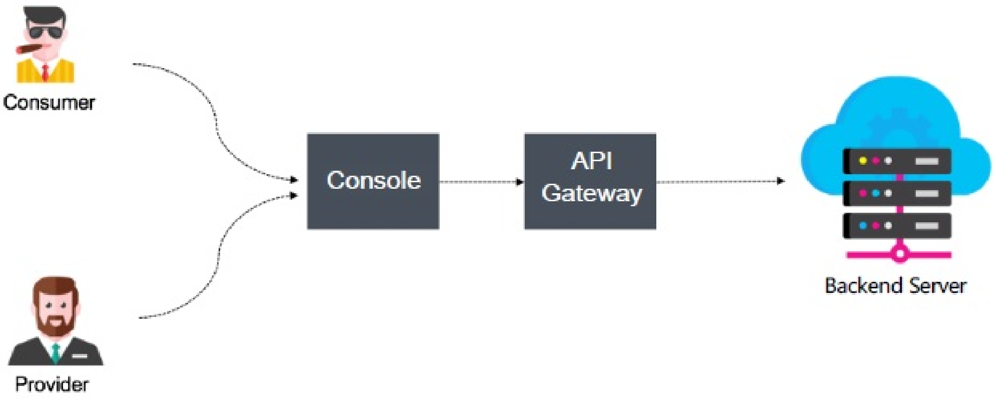


API Gateway allows you to publish an API group in three different environments: test, pre-release, and release environments. The test and pre-release environments are used by testers to test or debug APIs. The release environment is where the APIs can be used.

You can use the environment management function for API groups to set environment parameters for the test, pre-release, and online environments. The environment parameter is a common constant that can be customized for each environment. When you call an API, you can place the environment parameter in any location of the request. API Gateway identifies the environment based on the environment parameter in your request.

20.3.16. Online debugging

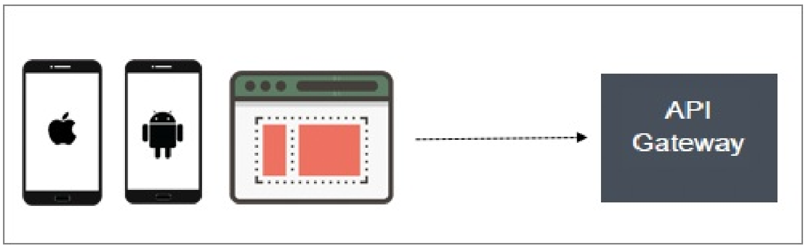
Online debugging



API Gateway provides the online API debugging function for system administrators and client developers.

20.3.17. Mock mode

Mock mode

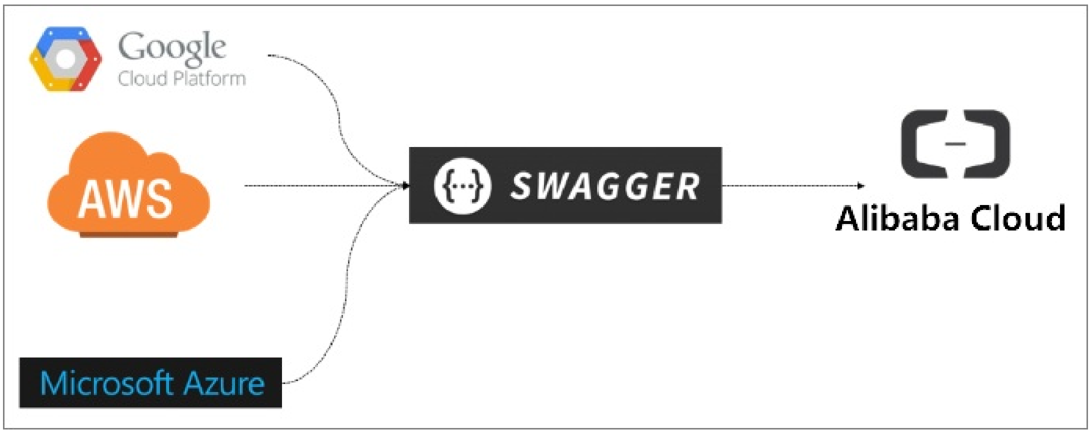


A project is typically developed by multiple partners working together toward a specific goal. The interdependence among various stakeholders often restricts individual members during the process, and misunderstandings may affect the development process or even delay the project schedule. You can mock expected responses to be returned to API callers during the project development process. This can greatly reduce misunderstanding among partners and greatly improve the development efficiency.

API Gateway supports the Mock mode.

20.3.18. Swagger file import

Swagger file import



Swagger is a widely-used specification to define and describe backend service APIs. You can create APIs in the API Gateway console by importing Swagger 2.0 files.

The API Gateway Swagger extension is based on Swagger 2.0. You can create the Swagger definition for API entities, and import the Swagger file to API Gateway for bulk creation or updating of API entities. API Gateway supports Swagger 2.0 by default, which is compatible with most Swagger specifications.

20.4. Benefits

- **Easy maintenance**

  After you register APIs in API Gateway, API Gateway handles all the API management issues such as documentation maintenance, version management of APIs, and SDK maintenance. This significantly reduces routine maintenance costs.

- **High performance**

  API Gateway maintains persistent connections between clients and API Gateway itself by supporting HTTP/2 and WebSocket. Both HTTP/2 and WebSocket are efficient binary protocols.

  API Gateway uses a distributed deployment and scales out automatically to handle a large number of API requests with low latency. API Gateway offers reliable and efficient features for your back-end services.

- **Stability**

  API Gateway has provided services to Alibaba Cloud public cloud users for over two years, and has a proven track record of performance. API Gateway provides stable services even in uncommon cases such as when over-sized packets are received, or when back-end services are unstable and slow to respond.

- **Security**

  API Gateway implements SSL encryption in the full link of communication to protect all data against eavesdropping during transmission.

  API Gateway implements signature verification in the full link of communication to prevent data tampering during transmission.

  API Gateway enables strict authorization management, anti-replay mechanism, parameter cleaning, IP address-based access control, and precise throttling.This ensures secure, stable and controllable services.

21.MaxCompute

21.1. What is MaxCompute?

21.1.1. Overview

MaxCompute is an offline data processing service developed by Alibaba Cloud based on the Apsara distributed operating system. It is capable of processing large amounts of data. MaxCompute can process terabytes or petabytes of data in scenarios where high real-time performance is not required. MaxCompute is used in fields such as log analysis, machine learning, data warehousing, data mining, and business intelligence.

MaxCompute is designed to provide an intuitive approach to analyze and process large amounts of data. You can analyze big data without having a deep knowledge of distributed computing. MaxCompute is widely used by Alibaba Group in scenarios such as data warehousing and BI analysis for large Internet enterprises, website log analysis, e-commerce transaction analysis, and exploration of user characteristics and interests.

MaxCompute provides the following features:

- Data channel
    - Tunnel: provides highly-concurrent offline data upload and download services. MaxCompute Tunnel enables you to upload or download large amounts of data to or from MaxCompute. You must use a Java API to access MaxCompute Tunnel.
    - DataHub: provides real-time upload and download services. Data uploaded by using DataHub is available immediately, while data uploaded by using MaxCompute Tunnel is not.
- Computing and analysis
    - SQL: MaxCompute stores data in tables and allows data queries by using SQL statements. MaxCompute can be used as traditional database software, but it is capable of processing terabytes and petabytes of data. MaxCompute SQL does not support transactions, indexes, or operations such as UPDATE and DELETE. The SQL syntax used in MaxCompute is different from that in Oracle and MySQL. SQL statements from other database engines cannot be seamlessly migrated to MaxCompute. MaxCompute SQL responds to queries within a few minutes or seconds, instead of milliseconds. MaxCompute SQL is easy to learn. You can get started with MaxCompute SQL based on your prior experience in database operations, without understanding distributed computing.
    - MapReduce: First proposed by Google, MapReduce is a distributed data processing model that has gained extensive attention and been used in a wide range of business scenarios. This document briefly describes the MapReduce model. You must have a basic knowledge of distributed computing and relevant programming experience before you use MapReduce. MapReduce provides a Java API.
    - Graph: an iterative graph computing framework provided by MaxCompute. Graph computing jobs use graphs to build models. A graph is a collection of vertices and edges that have values. MaxCompute Graph iteratively edits and evolves graphs to obtain analysis results.

  - Unstructured data access and processing in integrated computing scenarios: MaxCompute SQL cannot directly process external data, such as unstructured data from Object Storage Service (OSS). Data must be imported to MaxCompute tables by using relevant tools before computation. The MaxCompute team introduces the unstructured data processing framework to the MaxCompute system architecture to handle this issue.

    MaxCompute allows you to create external tables to process data from the following data sources:

    - Internal data sources: OSS, Tablestore, AnalyticDB, ApsaraDB RDS, Alibaba Cloud HDFS, and TDDL
    - External data sources: open source HDFS, MongoDB, and HBase
  - Unstructured data access and processing inside MaxCompute: MaxCompute allows you to read and write volumes. This enables MaxCompute to store and process unstructured data, which otherwise must be stored in an external storage system.
- Spark on MaxCompute: a big data analytics engine developed by Alibaba Cloud. It is used to provide big data processing capabilities for Alibaba, government agencies, and enterprises.
- Elasticsearch on MaxCompute: an enterprise-class system developed by Alibaba Cloud. It is used to retrieve information from large amounts of data and provide near-real-time search performance for government agencies and enterprises.

21.1.2. Features and benefits

Features

MaxCompute is a distributed system designed for big data processing. As one of the core services in the Alibaba Cloud computing solution, MaxCompute is used to store and compute structured data. It is also a basic computing component of the Alibaba Cloud big data platform. MaxCompute is designed to support multiple tenants and provide features such as data security and horizontal scaling. It provides a centralized graphical user interface (GUI) and centralized APIs for various data processing tasks of different users based on an abstract job processing framework. MaxCompute has the following features:

- Uses a distributed architecture that can be horizontally scaled based on your business requirements.
- Provides automatic storage and fault tolerance mechanisms to ensure high data reliability.
- Allows all computing tasks to run in sandboxes to ensure high data security.
- Uses RESTful APIs to provide services.
- Supports high-concurrent and high-throughput data uploads and downloads.
- Supports two types of service models: offline computing models and machine learning models.
- Supports data processing methods based on programming models such as SQL, MapReduce, Graph, and MPI.
- Supports multiple tenants, which allows multiple users to collaborate on data analysis.
- Manages user permissions based on access control lists (ACLs) and policies, which allows you to flexibly configure access control policies to prevent unauthorized data access.
- Supports Elasticsearch on MaxCompute, the enhanced Elasticsearch application.
- Supports Spark on MaxCompute, the enhanced Spark application.
- Supports the access to and processing of unstructured data.
- Supports the deployment of multiple clusters in a single region.

- Supports multi-region deployment.
- Uses the column store method and supports Key Management Service (KMS) to encrypt data files.
- Stores audit logs and automatically dumps them to a specific server directory for long-term storage and management.

Benefits

- **Excellent big data cloud service and real data sharing platform in China**: MaxCompute can be used for data warehousing, mining, analysis, and sharing. Alibaba Group uses this centralized data processing platform in several of its own services, such as Aliloan, Data Cube, DMP (Alimama), and Yu'e Bao.
- **Support for a large number of clusters, users, and concurrent jobs**: A single cluster can contain more than 10,000 servers and maintain 80% linear scalability. A single MaxCompute system supports more than 1 million servers in multiple clusters without limits. However, linear scalability is slightly affected. It also supports multi-data-center deployment in a zone. A single MaxCompute system supports more than 10,000 users, more than 1,000 projects, and more than 100 departments of multiple tenants. It can also support more than 1 million jobs (daily submitted jobs on average) and more than 20,000 concurrent jobs.
- **Big data computing at your fingertips**: You do not need to worry about the storage difficulties and prolonged computing processes caused by the increase of the data amount. MaxCompute automatically expands the storage and computing capabilities of clusters based on the data amount. This allows you to focus on data analysis and mining to maximize your data value.
- **Out-of-the-box service**: You do not need to worry about the creation, configuration, and O&M of clusters. Only a few simple steps are required to upload data, analyze data, and obtain analysis results in MaxCompute.
- **Secure and reliable data storage**: MaxCompute uses multi-level data storage and access control mechanisms to protect user data against loss, leaks, and interception. These mechanisms include multi-copy technology, read and write request authentication, and application and system sandboxes.
- **Reliable management nodes**: MaxCompute uses the multi-node cluster architecture. The management nodes of each component feature high availability. The faults that occur on O&M management nodes do not interrupt your services.
- **Powerful fault tolerance**: MaxCompute supports automatic fault tolerance for the failures of hard disks on servers in a cluster and supports hot swapping of hard disks. In the event of a hard disk failure, services can be restored within 2 minutes.
- **Comprehensive storage space management**: MaxCompute allows you to query information about both the storage capacity and usage of distributed file systems. It enables you to manage data lifecycles. MaxCompute also allows you to store data in different locations based on the data value or tag. For example, you can write temporary files to SSDs to accelerate I/O operations. This allows you to use cluster data more efficiently. MaxCompute also supports the self-optimizing Zstandard compression algorithm that provides the optimal compression ratio.
- **Comprehensive data backup**: MaxCompute allows you to perform full or incremental data backup and restore data from storage media. It also allows you to back up data for clusters in different data centers. This meets the requirements of mutual data backups among multiple data centers. You can use Apsara Big Data Manager (ABM) to manage the backup process in a visualized manner.
- **Secure and reliable access control**: MaxCompute allows you to manage data access permissions. The permissions include logon permissions, permissions to create tables, read and write permissions, and whitelist-related permissions. It also allows you to use the Apsara Uni-manager Management Console to manage administrative permissions, including administrator classification. You can use the

Apsara Uni-manager Management Console to manage user permissions in a centralized manner. You can view and manage the permission management features of all components in the system. You can also keep permission management details from common users and simplify permission management for administrators. This improves the usability and user experience of permission management.

- **Multi-tenancy for multi-user collaboration**: MaxCompute allows you to configure data access policies. This way, you can enable multiple data analysts in an organization to collaborate and make data accessible to users who are granted the required permissions. This ensures data security and maximizes productivity.
    - **Isolation**: You can submit the tasks of multiple tenants (projects) to different queues for concurrent running. Resources are isolated among tenants.
    - **Permission**: You can manage different tenants in a centralized manner and dynamically configure, manage, and isolate tenant resources. You can also collect statistics on the usage of tenant resources and manage multi-level tenants.
    - **Scheduling**: MaxCompute supports multi-tenant scheduling for multiple clusters an d resource pools.
- **Multi-region deployment**: You can specify compute clusters to efficiently use computing resources. Data exchanges between clusters are completed within MaxCompute, and data replication and synchronization between clusters are managed based on the configured policies. Therefore, cross-region data processing is no longer involved, which significantly reduces the waiting time for data processing.
- **Multi-device support**: You can use CPUs, hard disks, memory, and network interface controllers with different specifications in a single-component cluster to ensure maximum compatibility with existing devices. This applies only when cluster performance is not affected.

21.1.3. Benefits

Compared with traditional databases, MaxCompute has the following benefits.

Comparison of benefits

Benefit	Traditional databases	MaxCompute
System scalability	Disks cannot be shared across more than 100 nodes. Table and database sharding causes application data collision, resulting in massive computing overhead. This significantly compromises application analysis capabilities.	MaxCompute supports more than 10,000 nodes that can store more than 1.5 EBs of data. For example, during Alibaba's Double 11 event, MaxCompute processed more than 300 PBs of data in six hours.
Data type support	Cannot process unstructured data.	Can process both structured and unstructured data.
High availability	Redundant storage solutions are not available. Traditional backup and recovery approaches are inapplicable to large volumes of data (measured in PBs), and a single point of failure can cause the entire database to become unavailable.	Provides the shared-nothing architecture and multi-replica data model. This eliminates single points of failure.

Benefit	Traditional databases	MaxCompute
Complex computing capability	Iterative computing and graph computing capabilities are not available. The disk sharing technology and complex computing operations result in massive data exchanges between nodes, imposing tremendous bandwidth pressure.	Provides distributed storage and multiple computing frameworks such as MR, SQL, iterative computing, MPI, and graph computing.
Concurrency	A single large-scale computing task (such as index computing) can consume all system resources, and incur network and disk (data dictionary) bottlenecks. This makes highly concurrent access impossible.	Provides comprehensive multi-tenant isolation and resource management tools, so that you can easily view cluster resources and manage the resources used by each service. It can support up to 10,000 concurrent access requests.
Performance support	The indexing mechanism makes it difficult to support analytical applications of real-time data. Large amounts of data collision cause analytical predictions to take more than 24 hours, resulting in a performance bottleneck.	Focuses on the concurrent computing of large amounts of data. It provides available real-time data, and multiple high-performance computing capabilities, such as high-performance large-scale offline computing, real-time multi-dimensional analysis of large amounts of data, and stream computing.

21.1.4. Scenarios

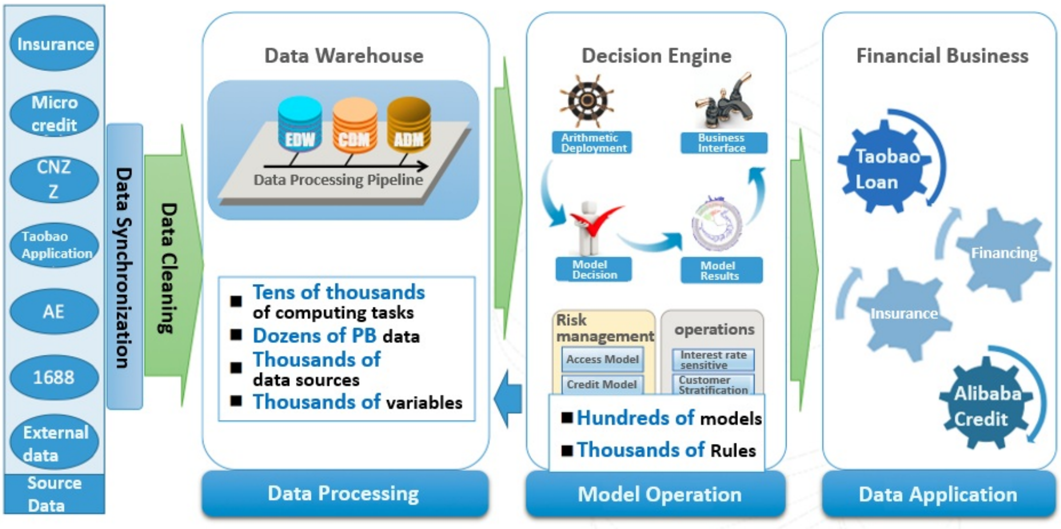
MaxCompute is designed for use in three big data processing scenarios:

- Establishment of SQL-based large data warehouses and BI systems
- Development of big data applications based on MapReduce and MPI distributed programming models
- Development of big data statistics models and data mining models based on statistics and machine learning algorithms

The following describe some real-world scenarios.

Data warehouse construction

Data warehouse construction



MaxCompute enables you to easily build a cloud-based data warehouse. With MaxCompute capabilities such as partitioning, data table statistics, and table life cycle management, you can easily enhance the storage of historical data warehouse information, divide hot and cold tables, and control data quality.

Alibaba's financial data warehousing team has built a sophisticated and powerful data warehousing system based on MaxCompute. This system provides six layers: the source data layer, ODS layer, enterprise data warehousing layer, common dimensional modeling layer, application marketplace layer, and presentation layer.

- The source data layer processes data from all sources, including Taobao, Alipay, B2B, and external data sources.
- ODS provides a temporary storage layer for data import.
- The enterprise data warehousing layer uses the 3NF modeling technique to divide data, including all historical data, by topic (such as item or shop).
- The common dimensional modeling layer uses the dimensional modeling approach to create modeling layers for general business applications. This layer shields the upper layers from changes in business requirements, and provides consistent and actionable data to the upper layers.
- The application marketplace layer is a demand-oriented layer that provides a data marketplace for specific applications.
- The presentation layer provides several data portals and services that can be accessed by applications.

This system architecture inevitably involves tasks such as metadata management.

The financial data warehouse is used to perform offline computing tasks based on MaxCompute SQL. It also uses a series of metric rules and algorithms to make decisions offline for online decision-making.

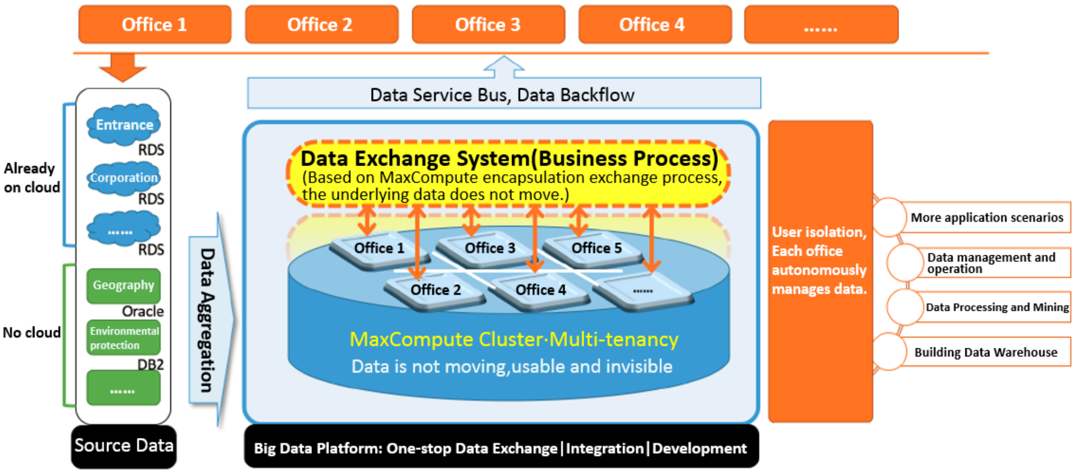
MaxCompute-based data warehouses differ from traditional databases in the following ways:

- **Historical data storage**: MaxCompute is able to store large amounts of data. You do not have to dump historical data to cheaper storage media as you would do in traditional databases.
- **Partitioning**: Traditional databases provide a wide range of partitioning methods such as range partitioning. MaxCompute provides fewer partitioning methods, but are sufficient for use in data warehousing scenarios. Whatever the method, you can build a data warehouse based on the same concept and principle as a table partition.
- **Wide tables**: MaxCompute stores data in fields, making it ideal for creating wide tables.

- **Data integration**: Traditional databases use stored procedures for data processing and integration. MaxCompute splits the logic of these operations into discrete SQL statements. Though the implementation is different, the algorithms are the same. In many years of experience, we found that splitting the operation logic into discrete SQL statements is clearer and more efficient, while stored procedures are more flexible and capable of processing complex logic.

Big data sharing and exchange

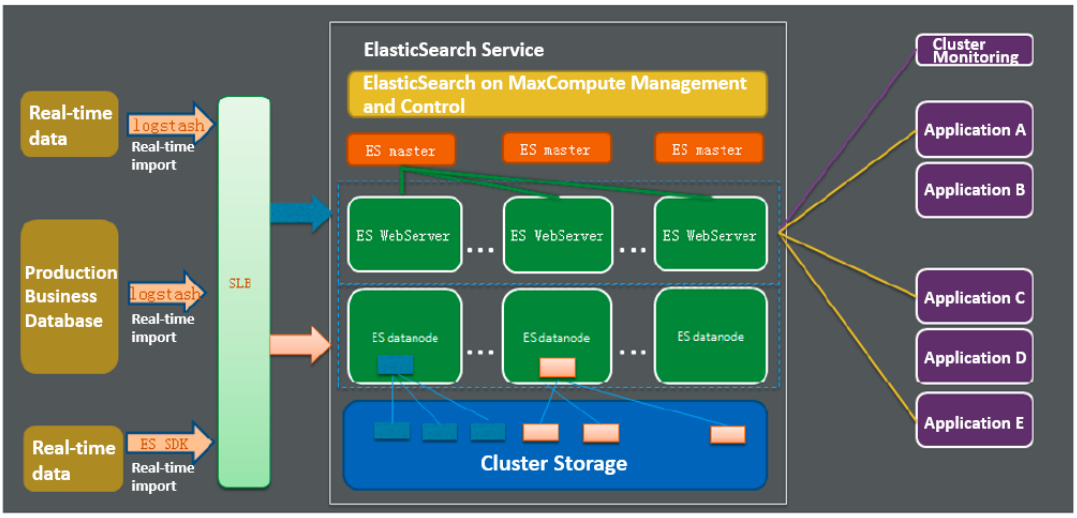
Big data sharing and exchange


Office 1
Office 2
Office 3
Office 4
......
Data Service Bus, Data Backflow
Already on cloud
No cloud
Entrance
RDS
Corporation
RDS
......
RDS
Geography
Oracle
Environmental protection
DB2
......
Source Data
Data Aggregation
Data Exchange System(Business Process)
(Based on MaxCompute encapsulation exchange process, the underlying data does not move.)
Office 1
Office 3
Office 5
Office 2
Office 4
......
MaxCompute Cluster·Multi-tenancy
Data is not moving,usable and invisible
Big Data Platform: One-stop Data Exchange|Integration|Development
User isolation, Each office autonomously manages data.
More application scenarios
Data management and operation
Data Processing and Mining
Building Data Warehouse

MaxCompute provides a wide range of permission management methods and flexible data access control policies. MaxCompute provides a wide range of access control mechanisms, including the ACL authorization, role-based authorization, policy authorization, cross-project authorization, and label security mechanism. MaxCompute provides column-level security solutions. This can meet the security requirements within an organization or across multiple organizations. For projects that demand high security, MaxCompute provides the project protection mechanism to prevent data leakage, and provides logs of all user operations to facilitate retrospective audits.

Typical applications of Elasticsearch on MaxCompute

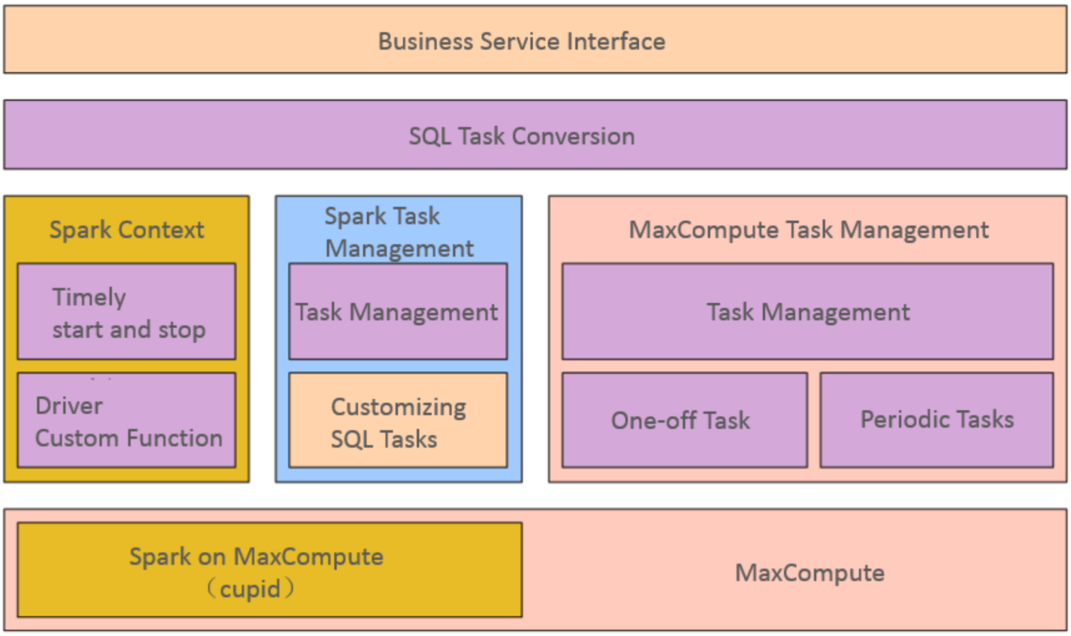
Typical applications



Elasticsearch on MaxCompute allows you to launch a set of Elasticsearch services by submitting jobs in a MaxCompute cluster. Native Elasticsearch code is not modified when applied in a project. **Elasticsearch on MaxCompute** runs in the same way as native Elasticsearch clusters.

Typical applications of Spark on MaxCompute

Typical applications



Spark on MaxCompute provides business computing platform and applications in Client mode. The preceding figure shows the application framework.

21.1.5. Service specifications

21.1.5.1. Software specifications

21.1.5.1.1. Overview

This section describes the software specifications of MaxCompute.

21.1.5.1.2. Control and service

Specifications

Item	Description
Number of control nodes	Greater than or equal to 3.
Number of MaxCompute front-end servers	Greater than or equal to 2. MaxCompute front-end servers can be deployed together with control nodes.
Number of tunnels	Greater than or equal to 2. Tunnels can be deployed together with compute nodes.

Item	Description
Number of DataHubs	Greater than or equal to 2. DataHubs can be deployed together with compute nodes.

21.1.5.1.3. Data storage

Specifications

Item	Description
Logical storage capacity per node	12 TB
Total storage capacity	The storage capacity can be scaled out by adding more nodes.

> ⓘ **Note** The size of logically stored data to a large extent determines the size of the cluster to be evaluated.

21.1.5.1.4. Size of a single cluster

Specifications

Item	Description
Offline computing cluster	An offline computing cluster can contain 3 to 10,000 machines.

21.1.5.1.5. Projects

Specifications

Item	Description
Creation of projects	Supported.
Acquisition of project metadata	Supported.
Deletion of projects	Supported.
Setting of the default lifecycle of tables	Supported.
Number of supported projects	Over 1,000

21.1.5.1.6. User management and security and access control

Specifications

Item	Description
Cross-project access	Supported. You can authorize cross-project access to organize tables and resources as packages and install them in other projects.
Service (odps_server and tunnel) authentication and access control	Supported. AccessKey ID and AccessKey Secret can be used to authenticate users and control their permissions.
Prevention of data outflow from a project	You can prevent data outflow and specify exceptions when necessary.
Label-based security	Label-based security (LabelSecurity) can be set to enable column-level access control. **Note** LabelSecurity is a mandatory access control policy that provides a wide range of security level settings.
Authorization to users	Supported.
Authorization to roles	Supported. You can customize roles and assign roles to users. Different roles are granted different permissions.
Project-specific authorization	The following permissions can be granted on a project: • View project information (excluding any project objects), such as the creation time. • Update project information (excluding any project objects), such as comments. • View the list of all object types in the project. • Create tables in the project. • Create instances in the project. • Create functions in the project. • Create resources in the project. • Create volumes in the project. • Grant all preceding permissions.

Item	Description
Table-specific authorization	The following permissions can be granted on a table: - Read table metadata. - Read table data. - Modify table metadata. - Overwrite or add table data. - Delete the table. - Grant all preceding permissions.
Function-specific authorization	The following permissions can be granted on a function: - Read. - Update. - Delete. - Grant all preceding permissions.
Authorization for resources, instances, jobs, and volumes	The following permissions can be granted on a resource, instance, job, or volume: - Read. - Update. - Delete. - Grant all preceding permissions.

Item	Description
Sandbox protection	The sandbox mechanism can restrict access to system resources in MapReduce and UDF programs. Specific restrictions are as follows: • Direct access to local files is not allowed. You can only read resource information and generate log information through System.out and System.err. **Note** You can view log information by running the Log command on the MaxCompute client. • Direct access to Apsara Distributed File System is not allowed. • JNI calls are not allowed. • Java threads cannot be created, and Linux commands cannot be executed by sub-threads. • Network access operations such as acquiring local IP addresses are not allowed. • Java reflection is not allowed. You cannot force access to protected or private members to be valid.
Control over the quotas of storage and computing resources	Supported. You can limit the number of files and used disk capacity in a project. You can also use quotas to limit the available CPU and memory capacity of the project.

21.1.5.1.7. Resource management and task scheduling

Specifications

Item	Description
File count quota and storage capacity quota	The quotas vary with projects.
Configuration of CPU quota for a resource group	You can configure the minimum or maximum number of virtual CPUs that can be used by a resource group.
Configuration of memory quota for a resource group	You can configure the minimum or maximum amount of virtual memory that can be used by a resource group.
Resource preemption	Preemption of resources within a quota group is supported.
Task scheduling methods	Fair scheduling and first-in-first-out (FIFO).

Item	Description
Configuration of task priorities	By default, task priorities are assigned in a project. You can configure the priorities as needed.
Restart of a failed task	Supported.
Speculative execution of a task	Supported.

21.1.5.1.8. Data tables

Specifications

Item	Description
Data storage methods	CFile data exclusive to MaxCompute is stored in columns in Apsara Distributed File System.
Data compression	Supported. The efficiency of compression is dependent on the data format. The compression ratio between the original and compressed data is 3:1. Infrequently accessed data can be archived in RAID to reduce the storage space it occupies by 50%.
Lifecycle	Supported.
Basic data types	BigInt, String, Boolean, Double, DateTime, and Decimal.
Partitions	Supported. Only String type partitions are supported.
Maximum number of columns	1,024
Maximum number of partitions	60,000
Partition levels	A table can contain up to five partition levels.
Views	Supported. A view can only contain one valid SELECT statement. Materialized views are not supported.
Statistics	Supported. You can define statistical metrics for data tables and view, analyze, and delete statistics.
Comments	Supported. You can make comments for both tables and columns. Comments can be up to 1024 characters in length.

21.1.5.1.9. SQL

21.1.5.1.9.1. DDL

Specifications

Item	Description
Creation of tables	Supported.
Deletion of tables	Supported.
Renaming of tables	Supported.
Creation of views	Supported.
Deletion of views	Supported.
Renaming of views	Supported.
Adding of partitions	Supported.
Deletion of partitions	Supported.
Adding of columns	Supported.
Modification of column names	Supported.
Modification of comments	Supported. You can modify comments for tables and columns.
Modification of the lifecycle of tables	Supported.
Disabling of the lifecycle for specific table partitions	Supported. The command syntax is as follows: `ALTER TABLE table_name [partition_spec] ENABLE|DISABLE LIFECYCLE`
Emptying of data from non-partitioned tables	Supported. The command syntax is as follows: `TRUNCATE TABLE table_name`
Modification of table owners	Supported.
Modification of the time when a table or partition was last modified	Supported.

21.1.5.1.9.2. DML

Specifications

Item	Description

Item	Description

Dynamic partition filtering	Supported. This technique can reduce the amount of data to be read. The command syntax is as follows:
	select_statment
	FROM from_statement
	WHERE PT1 IN (SUBQUERY) AND PT2 IN (SUBQUERY)... ;

Multiple outputs	Supported. A single SQL statement can contain up to 128 outputs. **Note** In each output, you can only specify once whether to target a partition in a partitioned table or target a non-partitioned table.
Data update and overwriting	Supported. Batch update is supported.
Aggregation	Supported.
Sorting	Supported. Sorting must be performed with the limit syntax.
Nested subqueries	Supported.
Joins	Supported. SQL joins include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.
UNION ALL	Supported.
CASE WHEN	Supported.
Relational operations	Supported.
Mathematical operations	Supported.
Logical operations	Supported.
Implicit conversions	Supported.

Item	Description
MAPJOIN	Supported. To speed the JOIN operation when volume of data is small, SQL loads all specified small tables into the memory of a program executing the JOIN operation. The default maximum data size is 512 MB. The maximum size cannot exceed 2 GB. Up to six small tables can be specified. **Note** Take note of the following limits: • The left table of a LEFT OUTER JOIN clause must be a large table. • The right table of a RIGHT OUTER JOIN clause must be a large table. • Both the left and right tables of an INNER JOIN clause can be large tables. • MAPJOIN cannot be used in a FULL OUTER JOIN clause. • MAPJOIN supports small tables as subqueries. • When MAPJOIN is used and a small table or subquery is referenced, you must reference the alias of the small table or subquery. • MAPJOIN supports both non-equivalent JOIN conditions and multiple conditions connected by using OR statements.
Query of the execution plans of DML statements	Supported. The description of the final execution plan corresponding to a DML statement can be displayed. The command syntax is as follows: `EXPLAIN <DML query>;`

21.1.5.1.9.3. Built-in functions

Specifications

Item	Description
Built-in functions	Supported. Built-in functions include string functions, date functions, mathematical functions, regular functions, and window functions.

21.1.5.1.9.4. User-defined functions

Specifications

Item	Description
Scalar functions	Supported. You can use the Java SDK and Python SDK to write scalar functions.

Item	Description
Aggregate functions	Supported. You can use the Java SDK and Python SDK to write aggregate functions.
Table functions	Supported. You can use the Java SDK and Python SDK to write table functions.
Implicit conversions	Supported.

21.1.5.1.10. MapReduce

21.1.5.1.10.1. Programming support

Specifications

Item	Description
Java language	Supported.
Standalone debugging mode	Supported.
Extended MapReduce model	Supported. A Map operation can be followed by any number of Reduce operations. Example: Map-Reduce-Reduce.

21.1.5.1.10.2. Job size

Specifications

Item	Description
Maximum number of mappers	100,000
Maximum number of reducers	2,000
Setting of the number of mappers and reducers	Supported. You can change the number of mappers by changing the input volume of each Map worker. By default, the number of reducers is set at 25% of the number of mappers. You can change this proportion to suit your business needs.
Setting of the memory of mappers and reducers	Supported. The default memory of a mapper or reducer is 2 GB.

Note The maximum numbers of mappers and reducers are related to the cluster size.

21.1.5.1.10.3. Input and output

Specifications

Item	Description
Input and output of a table	Supported.
Processing of unstructured data	Supported. Volumes are suited to store unstructured data. MaxCompute MapReduce can be used to process unstructured data.
Input and output of multiple tables	Supported. The numbers of inputs and outputs cannot exceed 128.
Reading of resources	Supported. A single task can reference up to 256 resources. The total size of all referenced resources cannot exceed 2 GB. ? **Note** The maximum number of read attempts for a resource is 64.

21.1.5.1.10.4. MapReduce computing

Specifications

Item	Description
Custom setup, map, and cleanup methods of mappers	Supported.
Custom setup, reduce, and cleanup methods of reducers	Supported. Transmitted messages are processed in the next iteration.
Custom partition columns or partitioners	Supported.
Configuration of mapper output columns to be sorted and grouped by keys	Supported. Note that custom key comparators are not supported.
Custom combiners	Supported.
Custom counters	Supported. A single job cannot have more than 64 custom counters.
Map-only jobs	Supported. To implement Map-only jobs, set the number of Reduce jobs to 0.
Configuration of job priorities	Supported.

21.1.5.1.11. Graph

21.1.5.1.11.1. Programming support

Specifications

Item	Description
Java language	Supported.
Standalone debugging mode	Supported.

21.1.5.1.11.2. Job size

Specifications

Item	Description
Maximum number of concurrent workers	1,000
Custom worker CPU and memory	Supported. By default, a worker has two CPU cores and 4 GB of memory. A worker can have up to eight CPU cores and 12 GB of memory.

21.1.5.1.11.3. Graph loading

Specifications

Item	Description
Loading of graph data from MaxCompute tables	Supported.
Division of graphs by vertex	Supported.
Custom partitioners	Supported.
Custom split size	Supported. The default split size is 64 MB.
Custom conflict logic upon data loading	Supported. For example, creating duplicate vertices and edges is considered a conflict logic.

21.1.5.1.11.4. Iterative computing

Specifications

Item	Description
Bulk Synchronous Parallel (BSP) computing model	Supported.
Transmission of messages between vertices	Supported. Transmitted messages are processed in the next iteration.

Item	Description
Multiple iteration termination conditions	1. The maximum number of iterations is reached. 2. All vertices enter the halted state. 3. An aggregator determines to terminate the iteration.
Automatic checkpoint mechanism	Supported.
Custom aggregators	Supported.
Custom combiners	Supported.
Custom counters	Supported. A single job cannot have more than 64 custom counters.
Custom conflict logic	Supported. For example, sending messages to a non-existent vertex is considered a conflict logic.
Writing of computing results to MaxCompute tables	Supported.
Configuration of job priorities	Supported.

21.1.5.1.12. Processing of unstructured data

21.1.5.1.12.1. Processing of Tablestore data

Specifications

Item	Description
Tablestore data types	All data types are supported.

21.1.5.1.12.2. Processing of OSS data

Specifications

Item	Description
User-defined split and range functions	Supported.
User-defined maximum number of concurrent mappers	Supported.
User-defined file list	Supported.

21.1.5.1.12.3. Multiple data sources

Specifications

Item	Description
Support for various open-source data formats through the STORED AS syntax	Supported data formats include PARQUET, ORC, SEQUENCEFILE, TEXTFILE, and AVRO.

21.1.5.1.13. Spark on MaxCompute

21.1.5.1.13.1. Programming support

Specifications

Item	Description
Native Apache Spark APIs	Supported. You can use native Spark APIs to write code and process data stored in MaxCompute.
Native methods to submit Spark jobs	Supported.
Multiple native Spark components	Spark SQL, Spark MLlib, GraphX, and Spark Streaming are currently supported.
Multiple programming languages	MaxCompute data can be processed using Scala, Python, Java, and R languages.

21.1.5.1.13.2. Data sources

Specifications

Item	Description
Processing of unstructured data	Supported. You can use Spark APIs to write code and process data stored in Object Storage Service (OSS) and Tablestore.
Processing of data from MaxCompute tables and resources	Supported.

21.1.5.1.13.3. Scalability

Specifications

Item	Description
Deep integration of Spark and MaxCompute	Supported. Spark and MaxCompute share cluster resources. Spark resources can be scaled from large-scale MaxCompute clusters.

21.1.5.1.14. Elasticsearch on MaxCompute

21.1.5.1.14.1. Programming support

Specifications

Item	Description
Native Elasticsearch APIs	Supported.

21.1.5.1.14.2. System capabilities

Specifications

Item	Description
Real-time analysis and retrieval of data at the petabyte level	Supported.
Web-based display for basic server metrics	Supported. A user-friendly O&M platform for index databases and full-text retrieval clusters can be used to monitor the status of index databases and machines in real time.
Data snapshot technology based on Apsara Distributed File System	Supported. Rapid data backup and recovery can be performed to ensure data reliability.
Millisecond-level response to keyword-based and comprehensive searches and second-level response to fuzzy searches	Supported.
Real-time analysis and retrieval of imported data and query response times within 500 milliseconds	Supported. The storage architecture is powered by the distributed cache-accelerated block device technology.
In-memory off-heap storage and processing of index data and fine-grained memory management	Supported.

21.1.5.1.15. Other extensions

The following extended plug-ins and tools are both client-specific and open-source. You can download the plug-ins and tools at https://github.com/aliyun/.

Specifications

Item	Description
R language support	RODPS is a plug-in for the MaxCompute client to support the R language.
Plug-ins and tools	Eclipse plug-ins and command line tools are available.
OGG	OGG plug-ins synchronize data from OGG to DataHub.

Item	Description
Flume	Flume plug-ins synchronize data from Flume to DataHub.
FluentD	FluentD plug-ins synchronize data from FluentD to DataHub.
JDBC	JDBC interfaces are partially supported.
Sqoop	Sqoop can be used to exchange data with MaxCompute.

21.1.5.2. Hardware specifications

The following table lists the hardware specifications of MaxCompute.

Hardware specifications

Node type	Server configuration	Number of nodes	Description
Management node	• CPU: dual-socket 8-core or higher • Memory: 256 GB or higher • Disk: two 4 TB NVMe U.2 SSDs • NIC: two 10 GE NICs for network bonding	N/A	We recommend that you use Intel Platinum 81xx series processors or higher configurations.

Node type	Server configuration	Number of nodes	Description
Control node	• CPU: dual-socket 8-core or higher • Memory: 128 GB or higher • Disk: one 4 TB SATA HDD with 7200 RPM performance • NIC: two 10 GE NICs for network bonding	8/13	• We recommend that you use Intel Platinum 81xx series processors or higher configurations. • When the number of data nodes is less than 500, the number of control nodes is 8. When the number of data nodes is more than 500, the number of control nodes is 13. • We recommend that you deploy data nodes in containers when the number of data nodes is less than 500. • When all control nodes are physical servers and the number of data nodes is less than 1,000, you can implement a hybrid deployment of control nodes and data nodes based on your actual needs. • The system disk capacity is greater than or equal to 240 GB.
Hybrid deployment of management nodes and control nodes	• CPU: dual-socket 8-core or higher • Memory: 256 GB or higher • Disk: one 4 TB NVMe U.2 SSD • NIC: two 10 GE NICs for network bonding	N/A	• Hybrid deployment is recommended when the number of data nodes is less than 500 and is not expected to be increased. • Assume that the number of data nodes is approximately 500 and is expected to increase to more than 500. When you deploy the nodes for the first time, we recommend that you deploy them separately on physical servers.

Node type	Server configuration	Number of nodes	Description
Data node	- CPU: dual-socket 8-core or higher - Memory: 128 GB or higher - Disk: twelve 2 TB, 4 TB, 6 TB, or 8 TB SATA HDDs with 7200 RPM performance - NIC: two 10 GE NICs for network bonding	Depends on the amount of data.	- We recommend that you use Intel Golden 61xx series processors or higher configurations. - The recommended ratio of core quantity to memory capacity is 1:4. - We recommend that you add a 4 TB NVMe U.2 SSD when the number of cores is greater than or equal to 48. - Number of nodes = [(Total planned data volume × Data expanding rate (1.3) × Data compression rate (1) × Number of replicas (3))/Disk utilization rate (0.85)/Disk formatting loss (0.9)/((Number of disks (12) - Number of system reserved blocks (1)) × Disk capacity (8 TB))] rounded up.

ⓘ Note

- We recommend that you use the preceding configurations in offline scenarios as needed.
- We recommend that you do not use two or more machine types for compute nodes of MaxCompute.
- We recommend that you do not use both 1 GE and 10 GE NICs for MaxCompute.
- The configuration of machines to be added cannot be lower than that of the existing machines.
- The reuse of compute nodes needs to be evaluated together with the business side.

21.1.5.3. Specifications of DNS resources

Specifications

Resource name	Domain name	Description
	odps_frontend_server_inner_dns	The internal domain name of the MaxCompute front-end server. This domain name is not subject to VPC.

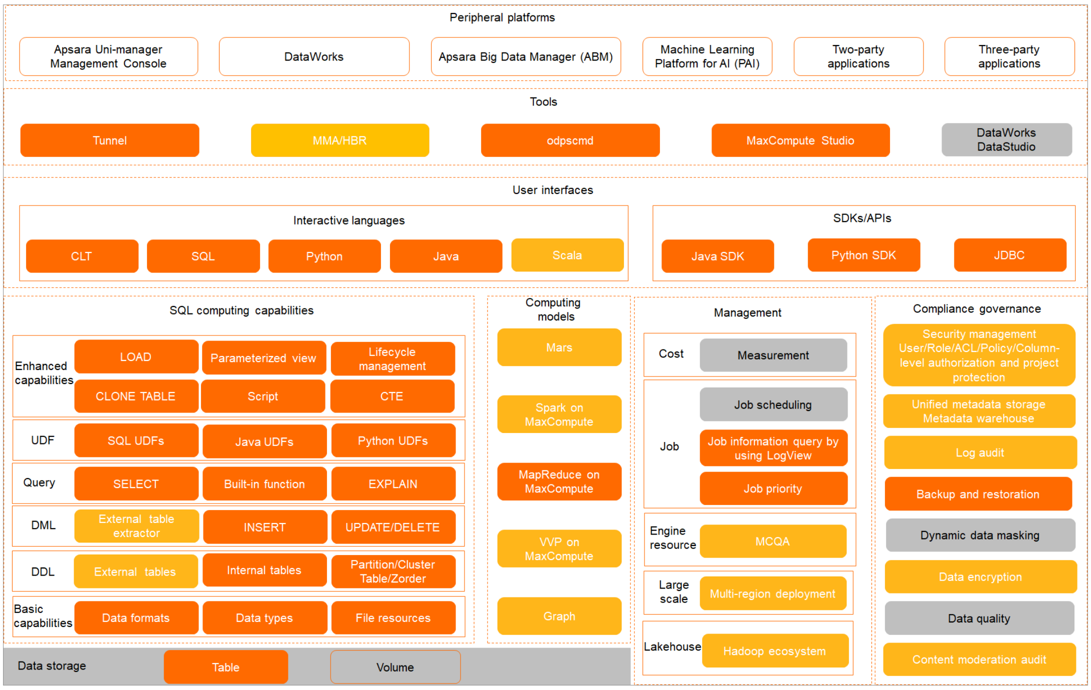
Resource name	Domain name	Description
odps_frontend	odps_frontend_server_public_dns	The private domain name of the MaxCompute front-end server.
	odps_frontend_server_internet_dns	The public domain name of the MaxCompute front-end server.
tunnel_frontend	odps_tunnel_frontend_server_inner_vip	The internal domain name of the front-end server for MaxCompute Tunnel. This domain name is not subject to VPC.
	odps_tunnel_frontend_server_public_vip	The private domain name of the front-end server for MaxCompute Tunnel.
	odps_tunnel_frontend_server_internet_vip	The public domain name of the front-end server for MaxCompute Tunnel.
cupid_web_proxy	odps_jobview_dns	The internal domain name of the MaxCompute Jobview. This domain name is not subject to VPC.
logview	odps_logview_inner_dns	The internal domain name of the MaxCompute Logview. This domain name is not subject to VPC.
	odps_logview_public_dns	The private domain name of the MaxCompute Logview.
web_console	odps_webconsole_inner_dns	The internal domain name of the MaxCompute Web console. This domain name is not subject to VPC.
	odps_webconsole_public_dns	The private domain name of the MaxCompute Web console.

21.2. Architecture

This topic describes the architecture of MaxCompute. The architecture and descriptions are for reference only and subject to the released product type and supplementary features.

The following figure shows the architecture of MaxCompute.

Architecture



🟧 indicates the basic features of MaxCompute. 🟨 indicates the enhanced features of MaxCompute. ⬜ indicates the features provided by external systems.

Category	Description
Peripheral platforms	MaxCompute supports the following peripheral platforms: • Apsara Uni-manager Management Console: a unified and intelligent O&M platform. For more information, see *Apsara Uni-manager Management Console User Guide*. • DataWorks: a visualization tool. You can use DataWorks to perform common operations, such as synchronize data, schedule jobs, and generate reports. For more information, see *DataWorks Technical White Paper*. • Apsara Big Data Manager (ABM): provides an easy method for field engineers to manage MaxCompute. For more information, see *Apsara Big Data Manager Technical White Paper*. • Machine Learning Platform for AI (PAI): a machine learning algorithm platform based on MaxCompute. For more information, see *Machine Learning Platform for AI Technical White Paper*. • Two-party applications: other Alibaba Cloud services supported by MaxCompute, such as DataV. • Three-party applications: other services that are compatible with MaxCompute.

Category	Description
Tools	MaxCompute supports the following tools: - Tunnel: a tunnel service. MaxCompute allows you to import heterogeneous data into or export the data from MaxCompute by using Tunnel. For more information, see *Tunnel* in *MaxCompute Product Introduction*. - MaxCompute Migration Assist (MMA): the data migration tool of MaxCompute. If you use MMA, Meta Carrier is used to access your Hive metastore service and capture Hive metadata. Then, MMA uses the Hive metadata to generate data definition language (DDL) statements and SQL statements of Hive user-defined table-valued functions (UDTFs). The DDL statements are used to create MaxCompute tables and their partitions. The SQL statements of Hive UDTFs are used to migrate data. - Hybrid backup recovery (HBR): integrates data backup and migration capabilities of Apsara Stack. - odpscmd: the MaxCompute client. For more information, see *Client* in *MaxCompute User Guide*. - MaxCompute Studio: the big data integrated development environment tool that is provided by MaxCompute. MaxCompute Studio is installed on a developer client. It is a development plug-in that Alibaba Cloud provides for the popular integrated development environment (IDE) IntelliJ IDEA. - DataWorks DataStudio: a visualized development platform provided by DataWorks. For more information, see *DataWorks User Guide*.
User interfaces	MaxCompute supports the following interfaces: - Interactive languages: CLI, SQL, Python, Java, and Scala. - SDKs and APIs: SDK for Java, SDK for Python, and Java Database Connectivity (JDBC). For more information, see *MaxCompute Developer Guide*.
SQL computing capabilities	MaxCompute supports the following SQL computing capabilities: - Enhanced capabilities: support LOAD, parameterized view, lifecycle management, and CLONE TABLE. - User-defined functions (UDFs): include SQL UDFs, Java UDFs, and Python UDFs. - Query: the query operations, such as SELECT and EXPLAIN statements and built-in functions. - Data manipulation language (DML) statements: include INSERT, UPDATE, and DELETE. - DDL statements: allow you to create internal tables, external tables, clustered tables, and partitioned tables. - Basic capabilities: support multiple data types and data formats and allow you to upload resource files. For more information, see *MaxCompute SQL* in *MaxCompute User Guide*.

Category	Description
Computing models	MaxCompute supports the following computing models: - Mars: a tensor-based unified distributed computing framework. Mars can use parallel and distributed computing technologies to accelerate data processing for Python data science stacks. For more information, see *Mars* in *MaxCompute User Guide*. - Spark on MaxCompute: a solution developed by Alibaba Cloud to enable the seamless use of Spark on the MaxCompute platform. It supplements a wide variety of features to MaxCompute. For more information, see *Spark on MaxCompute* in *MaxCompute User Guide*. - MapReduce on MaxCompute: allows you to run MapReduce jobs on MaxCompute. For more information, see *MaxCompute MapReduce* in *MaxCompute User Guide*. - VVP on MaxCompute: encapsulates the features of Realtime Compute for Apache Flink that is developed on the Ververica Platform (VVP) based on MaxCompute resources. You can use the Cupid joint computing platform to complete the operations related to real-time computing by using the underlying storage and computing resources of MaxCompute on the VVP UI. For more information, see *VVP On MaxCompute* in *MaxCompute User Guide*. - Graph: a processing framework designed for iterative graph computing. For more information, see *MaxCompute Graph* in *MaxCompute User Guide*.
Management	MaxCompute can be managed from the following aspects: - Cost: measures resource usage. - Job: provides mechanisms to manage jobs. For example, you can use these mechanisms to schedule jobs, use LogView to view job information, and set job priorities. - Engine resource: supports high-performance MaxCompute Query Acceleration (MCQA). - Large scale: allows you to deploy MaxCompute clusters across regions. - Lakehouse: a data management platform that combines data lakes and data warehouses. It integrates the flexibility and diverse ecosystems of data lakes with the enterprise-class deployment of data warehouses. For more information, see *MaxCompute Operations and Maintenance Guide*.

Category	Description
Compliance governance	MaxCompute allows you to use the following methods for compliance governance: • Security management: allows you to control the permissions of users and roles, and supports multiple authorization methods, such as ACL-based, policy-based, and column-level authorization. • Unified metadata storage: stores metadata in a centralized manner. • Log audit: audits different log data of different users. • Backup and restoration: allows you to back up and restore data from a storage system. • Dynamic data masking: allows you to query data masking rules in DataWorks. • Data encryption: uses Key Management Service (KMS) to encrypt data for storage. This way, MaxCompute can provide static data protection to meet the requirements of enterprise governance and security compliance. • Data quality: DataWorks provides an end-to-end platform that supports quality verification, notification, and management services for various heterogeneous data sources. • Content moderation audit: uses the content moderation engine to identify and audit pornographic, violent, and illegal content. For more information about security, see *MaxCompute Security White Paper*.
Data storage	MaxCompute stores data as tables or volumes.

21.3. Features

21.3.1. Tunnel

21.3.1.1. Overview

Data upload and download tools provided by MaxCompute are developed based on Tunnel SDK. This topic describes the major APIs of Tunnel SDK.

The usage of the SDK varies based on its version. For more information, see SDK Java Doc.

API	Description
TableTunnel	The entry class that is used to access MaxCompute Tunnel.
TableTunnel.UploadSession	A session that uploads data to a MaxCompute table.
TableTunnel.DownloadSession	A session that downloads data from a MaxCompute table.
InstanceTunnel	The entry class that is used to access MaxCompute Tunnel.

API	Description
InstanceTunnel.DownloadSession	A session that downloads data from a MaxCompute SQL instance. The SQL instance must start with the SELECT keyword and is used to query data.

Note

- If you use Maven, you can search for odps-sdk-core in the Maven repository to find the latest version of the SDK for Java. You can configure the Maven dependency in the following way:

  ```
  <dependency>
    <groupId>com.aliyun.odps</groupId>
    <artifactId>odps-sdk-core</artifactId>
    <version>0.36.2</version>
  </dependency>
  ```

- The endpoint of MaxCompute Tunnel supports automatic routing based on the MaxCompute endpoint settings.

21.3.1.2. TableTunnel

This topic describes the TableTunnel API.

TableTunnel is an entry class of the MaxCompute Tunnel service. You can use TableTunnel to upload or download only table data. Views cannot be uploaded or downloaded.

Definition

The following code defines the TableTunnel API.

```
public class TableTunnel {
public DownloadSession createDownloadSession(String projectName, String tableName);
public DownloadSession createDownloadSession(String projectName, String tableName, PartitionSpec par
titionSpe c);
public UploadSession createUploadSession(String projectName, String tableName);
public UploadSession createUploadSession(String projectName, String tableName, PartitionSpec partitionS
pec);
public DownloadSession getDownloadSession(String projectName, String tableName, PartitionSpec partiti
onSpec, String id);
public DownloadSession getDownloadSession(String projectName, String tableName, String id);
public UploadSession getUploadSession(String projectName, String tableName, PartitionSpec partitionSpe
c, String id);
public UploadSession getUploadSession(String projectName, String tableName, String id); public void setEn
dpoint(String endpoint);
}
```

Description

- The lifecycle of a TableTunnel instance starts from the time it is created to the time data upload or download is complete.
- TableTunnel provides a method to create UploadSession and DownloadSession objects.

TableTunnel.UploadSession is used to upload data. TableTunnel.DownloadSession is used to download data.

- A session refers to the process of uploading or downloading a table or partition. A session consists of one or more HTTP requests to Tunnel RESTful APIs.
- In an upload session, each RecordWriter matches an HTTP request and is identified by a unique block ID. The block ID is the name of the file that corresponds to the RecordWriter.
- If you use the same block ID to enable a RecordWriter multiple times in the same session, the data uploaded after the RecordWriter calls the close() method for the last time overwrites all the data that is previously uploaded. This feature can be used to retransmit a data block that fails to be uploaded.
- In UploadSession of TableTunnel:
  - If the boolean overwrite parameter is not specified, the INSERT INTO statement is used.
  - If the boolean overwrite parameter is set to True, the INSERT OVERWRITE statement is used.
  - If the boolean overwrite parameter is set to False, the INSERT INTO statement is used.

  Descriptions of INSERT OVERWRITE and INSERT INTO:

  - INSERT INTO: Upload sessions of the same table or partition do not affect each other. Data uploaded in each session is saved in different directories.
  - INSERT OVERWRITE: All data in a table or partition is overwritten by the data in the current upload session. If you use this statement to upload data, do not perform concurrent operations on the same table or partition.

Implementation process

1. The RecordWriter.write() method uploads your data as files to a temporary directory.
2. The RecordWriter.close() method moves the files from the temporary directory to a data directory.
3. The session.commit() method moves all files from the data directory to the directory in which the required table is saved, and updates the table metadata. This way, the data moved to a table in the current job is visible to other MaxCompute jobs such as SQL and MapReduce jobs.

Limits

- The value of a block ID must be greater than or equal to 0 but less than 20000. The size of the data that can be uploaded in a block cannot exceed 100 GB.
- A session is uniquely identified by its ID. The lifecycle of a session is 24 hours. If your session times out because large amounts of data are transmitted, you must transmit your data in multiple sessions.
- The lifecycle of an HTTP request that corresponds to a RecordWriter is 120 seconds. If no data flows over an HTTP connection within 120 seconds, the server closes the connection.

  > **Note** HTTP has an 8 KB buffer. When you call the RecordWriter.write() method, your data may be saved to the buffer and no inbound traffic flows over the HTTP connection. In this case, you can call the TunnelRecordWriter.flush() method to forcibly flush data out of the buffer.

- If you use a RecordWriter to write logs to MaxCompute, the write operation may time out due to unexpected traffic fluctuations. To avoid such issues, take note of the following points:
  - We recommend that you do not use a RecordWriter for each data record. If you use a RecordWriter for each data record, a large number of small files are generated, because each RecordWriter corresponds to a file. This affects the performance of MaxCompute.

  - If the size of cached code reaches 64 MB, we recommend that you use a RecordWriter to write multiple data records at the same time.

- The lifecycle of a RecordReader is 300 seconds.

21.3.1.3. InstanceTunnel

This topic describes the InstanceTunnel API.

InstanceTunnel is an entry class to access the MaxCompute Tunnel service. You can use InstanceTunnel to download the execution results of an SQL instance that executes a SELECT statement.

Definition

The following code shows the definition of the InstanceTunnel API:

```
public class InstanceTunnel{
 public DownloadSession createDownloadSession(String projectName, String instanceID);
 public DownloadSession createDownloadSession(String projectName, String instanceID, boolean limitEnab
led);
 public DownloadSession getDownloadSession(String projectName, String id);
 }
```

Parameters:

- projectName: the name of a project.
- instanceID: the ID of an instance.

Limits

InstanceTunnel provides an easy way to obtain instance execution results. However, it is subject to the following permission limits to ensure data security:

- If the number of data records is less than or equal to 10,000, all users who have read permissions on the specified instance can use InstanceTunnel to download the data records. The same rule applies to data queries by calling a RESTful API.
- If the number of data records is greater than 10,000, only users who have the read permissions on all the source tables from which the specified instance queries data can use InstanceTunnel to download the data records.

21.3.1.4. UploadSession

This topic describes the UploadSession API.

This API is used to upload data to MaxCompute tables.

Definition

The following code defines the UploadSession API:

```
public class UploadSession {
  UploadSession(Configuration conf, String projectName, String tableName,
         String partitionSpec) throws TunnelException;
  UploadSession(Configuration conf, String projectName, String tableName,
         String partitionSpec, String uploadId) throws TunnelException;
  public void commit(Long[] blocks);
  public Long[] getBlockList();
  public String getId();
  public TableSchema getSchema();
  public UploadSession.Status getStatus();
  public Record newRecord();
  public RecordWriter openRecordWriter(long blockId);
  public RecordWriter openRecordWriter(long blockId, boolean compress);
  public RecordWriter openBufferedWriter();
  public RecordWriter openBufferedWriter(boolean compress);
}
```

Notice

- Block IDs that are used within the same upload session must be unique. After you use a block ID to enable RecordWriter, write multiple data records at the same time, call close, and then call commit to complete data upload in an upload session, you cannot use this block ID to enable another RecordWriter to write data.
- The maximum size of a block is 100 GB. We recommend that the volume of data written to each block be greater than 64 MB. Otherwise, the computing performance deteriorates significantly.
- The lifecycle of a session on the server is 24 hours.
- When you upload data, a network action is triggered each time RecordWriter writes 8 KB of data. If no network actions are triggered within 120 seconds, the server closes the connection and RecordWriter becomes unavailable. You must enable a new RecordWriter to write data.
- We recommend that you use the openBufferedWriter operation to upload data. This operation does not show the blockId value but contains an internal data cache. If a block fails to be uploaded, this operation automatically retries the upload process.
- The overwrite mode is added by using the commit method. You can use the overwrite mode to submit data. If you submit data in overwrite mode, the submitted data overwrites the existing data in the table or partition.

  Notice Undefined behavior occurs when you submit data in overwrite mode in multiple concurrent sessions. This may affect data accuracy. To avoid this issue, you must determine the number of concurrent sessions in which you submit data in overwrite mode.

Description

- Lifecycle: indicates the period from the time an upload instance is created to the time data is uploaded.
- Upload instance. You can call the Constructor method or use TableTunnel to create an upload

instance.

  - Request mode: synchronous.
  - The server creates a session for the upload instance and generates a unique upload ID to identify the upload instance. You can run the `getId` command on the client to obtain the upload ID.

- Data upload
  - Request mode: synchronous.
  - Call the openBufferedWriter operation to generate a RecordWriter instance. The blockId parameter identifies the data that is uploaded this time and describes the data position in the whole table. The value of blockId is in the range of [0,20000]. If the upload fails, you can upload the data again based on the block ID.

- Upload status
  - Request mode: synchronous.
  - Call the `getStatus` method to obtain the current upload status.
  - Call the `getBlockList` method to obtain the blocks that are uploaded. Compare the result with the list of block IDs that were previously sent to the server and re-upload the blocks that fail to be uploaded.

- Upload termination
  - Request mode: synchronous.
  - Call the Commit(Long[] blocks) method. The blocks parameter indicates the blocks that are uploaded. The server verifies the block list.
  - Verification enhances data accuracy. If the provided list of block IDs is different from the list on the server, an error is returned.
  - If the commit operation fails, try again.

- State description
  - UNKNOWN: This is the initial state when the server creates a session.
  - NORMAL: The upload session is created.
  - CLOSING: When you call the complete method to end an upload session, the server changes the state to CLOSING.
  - CLOSED: The data upload is complete. The data is moved to the directory where the result table is saved.
  - EXPIRED: The upload session times out.
  - CRITICAL: An error occurs.

21.3.1.5. DownloadSession

This topic describes the DownloadSession API.

This API is used to download data from MaxCompute tables.

Definition

The following code defines the DownloadSession API:

```
public class DownloadSession {
  DownloadSession(Configuration conf, String projectName, String tableName,
          String partitionSpec) throws TunnelException
  DownloadSession(Configuration conf, String projectName, String tableName,
          String partitionSpec, String downloadId) throws TunnelException
  public String getId()
  public long getRecordCount()
  public TableSchema getSchema()
  public TableTunnel.DownloadStatus getStatus()
  public RecordReader openRecordReader(long start, long count)
  public RecordReader openRecordReader(long start, long count, boolean compress)
}
```

Description

- Lifecycle: indicates the period from the time a download instance is created to the time data is downloaded.
- Download instance: You can call the constructor method or use TableTunnel to create a download instance.
  - Request mode: synchronous.
  - The server creates a session for the download instance and generates a unique download ID to identify the download instance. You can call the `getId` method on the client to obtain the download ID.
  - This operation results in high overheads. The server creates indexes for data files. If a large number of data files exists, it takes a long time to create indexes for the data files.
  - The server returns the total number of records. You can start multiple download sessions at the same time to download data based on the total number of data records.
- Data download
  - Request mode: asynchronous.
  - Call the openRecordReader API to generate a RecordReader instance. The start parameter identifies the start position of the data record in this download session. The value of this parameter starts from 0 and must be greater than or equal to 0. The count parameter identifies the number of data records that are downloaded in this session. The value of the count parameter must be greater than 0.
- Download status
  - Request mode: synchronous.
  - Call the `getStatus` method to obtain the download status.
- Status description
  - UNKNOWN: This is the initial state when the server creates a download session.
  - NORMAL: The download object is created.
  - CLOSED: The download is complete.
  - EXPIRED: The download session times out.

21.3.1.6. TunnelBufferedWriter

This topic describes the TunnelBufferedWriter API.

This API is used to upload data.

The upload process is complex due to limits on block management and connection timeout on the server. The Tunnel SDK provides an enhanced RecordWriter of TunnelBufferWriter to simplify the upload process.

Definition

The following code defines the TunnelBufferedWriter API:

```
public class TunnelBufferedWriter implements RecordWriter {
  public TunnelBufferedWriter(TableTunnel.UploadSession session, CompressOption option) throws IOExc
eption;
  public long getTotalBytes();
  public void setBufferSize(long bufferSize);
  public void setRetryStrategy(RetryStrategy strategy);
  public void write(Record r) throws IOException;
  public void close() throws IOException;
}
```

Description

- Lifecycle: indicates the period from the time RecordWriter is created to the time data is upload.
- TunnelBufferedWriter instance: You can call openBufferedWriter of UploadSession to create a TunnelBufferedWriter instance.
- Data upload: If you call Write, data records are first written to the local cache. After the cache is full, multiple data records are submitted to the server at a time to avoid a connection timeout. If data upload fails, the system automatically retries the upload operation.
- Upload termination: You can call close and then commit of UploadSession to terminate the upload process.
- Buffer control: You can call setBufferSize to change the memory (in bytes) occupied by the buffer. We recommend that you set the memory size to a value greater than or equal to 64 MB. This prevents excessive small files from being generated on the server, which may affect the processing performance. The value ranges from 1 MB to 1000 MB. The default value is 64 MB.
- Retry policy settings: The following policies are provided: EXPONENTIAL_BACKOFF, LINEAR_BACKOFF, and CONSTANT_BACKOFF. The following code snippet sets the number of Write retries to 6. To avoid unnecessary retries, you can perform each retry after an exponential interval, such as 4s, 8s, 16s, 32s, 64s, and 128s. By default, the interval starts from 4s.

  ```
  RetryStrategy retry
   = new RetryStrategy(6, 4, RetryStrategy.BackoffStrategy.EXPONENTIAL_BACKOFF)
  writer = (TunnelBufferedWriter) uploadSession.openBufferedWriter();
  writer.setRetryStrategy(retry);
  ```

  > **Note** We recommend that you retain the preceding settings.

21.3.2. SQL

MaxCompute SQL is a structured query language whose syntax is similar to Oracle, MySQL, and Hive SQL. MaxCompute SQL can be regarded as a subset of standard SQL. However, MaxCompute SQL is not equivalent to a database, because it does not possess many characteristics that a database has, such as transactions, primary key constraints, and indexes.

MaxCompute SQL is applicable to scenarios that have large amounts of data (measured in TBs) and that do not have high real-time processing requirements. It takes a relatively long time to prepare and submit each job. Therefore, MaxCompute SQL is not optimal for services that need to process thousands of transactions per second.

21.3.3. MapReduce

MapReduce is a programming model equivalent to Hadoop MapReduce. This model is used for parallel MaxCompute operations on TB-level large-scale datasets.

You can use the MapReduce Java API to write MapReduce programs to process MaxCompute data. The Map and Reduce concepts are borrowed from functional and vector programming languages. This helps programmers run their programs on distributed systems without having to perform distributed parallel programming.

MapReduce works only after you specify a Map function and a concurrent Reduce function. The Map function maps a group of key-value pairs to another group of key-value pairs. The Reduce function ensures that all elements in the mapped key-value pairs share the same key group.

MaxCompute MapReduce has the following characteristics:

- Provides Hadoop-style MapReduce functions designed for MaxCompute (used to process tables and volumes).
- Supports the input and output of only built-in data types of MaxCompute.
- Supports the input and output of multiple tables to different partitions.
- Capable of reading resources.
- Does not allow you to use views as data inputs.
- Provides a limited sandbox security environment.

The following procedure shows how MapReduce processes data:

1. Before you perform Map operations, ensure that partition is set for the input data. The input data is divided into equally sized blocks called partitions. Each partition is processed as the input of a single Map worker so that multiple Map workers can work in parallel.
2. After partitioning, multiple Map workers start processing the data in parallel. Each Map worker reads its respective partition data, computes the data, and exports the result to Reduce.

   > **Note** When a Map worker generates data, it must specify a key for each output record. The key determines the Reduce worker for which the data entry is targeted. Multiple keys may correspond to a single Reduce worker. Data entries with the same key are sent to the same Reduce worker. A single Reduce worker may receive data entries for multiple keys.

3. Before entering the Reduce stage, the MapReduce framework sorts data based on Key values to make data entries with the same Key value adjacent. If you specify Combiner, the framework will call Combiner to combine data entries that share the same Key value.

> Note You can customize the Combiner logic. Unlike the typical MapReduce framework protocol, MaxCompute requires the input and output parameters of Combiner to be consistent with those of Reduce. This process is generally called Shuffle.

4. When entering the Reduce stage, data entries with the same Key value will be in the same Reduce worker. A single Reduce worker may receive data from multiple Map workers. Each Reduce worker performs the Reduce operation on multiple data entries with the same Key value. After the Reduce operation, all data of the same key is converted into a single value.

> Note This topic only provides a brief introduction to MapReduce. For more information, see related documentation.

21.3.4. Graph

Graph is the computing framework of MaxCompute designed for iterative graph processing. It provides programming interfaces similar to Pregel, allowing you to develop efficient machine learning and data mining algorithms.

Large amounts of data on the Internet is structured as graphs, such as social networking and logistics information. Graph computing models are iterative computing models. Throughout the entire computing process, multiple iterations are performed to achieve convergence. For example, for machine learning algorithms that require iterative learning model parameters, Graph is more suited than MapReduce. In common usage scenarios, you can abstract a question as a graph. Then, you can set the vertex as the center of the graph, and use supersteps for iterative updating.

MaxCompute Graph currently works in two modes:

- Offline mode: suitable for large-scale computing. Similar to MapReduce jobs, this mode involves loading and computing.
- Interactive mode: suitable for small-scale computing. You can implement a UDF and then use the command line for interaction.

In offline mode, loading and computing are independent processes. Loaded data resides in the memory. You can apply different computing logics to the loaded data. For example, the risk control department may load a set of data once a day. The operations personnel will apply different query logics to the data to view the relationships between the data.

MaxCompute Graph has been applied to many businesses in Alibaba. For example, weighted PageRank algorithms are used to compute influence metrics for Alipay users, and variational Bayesian EM models are used to predict users' car brands based on the properties of the items purchased by users.

21.3.5. Unstructured data processing in integrated computing scenarios

Alibaba Cloud introduced the MaxCompute-based unstructured data processing framework so that MaxCompute SQL can directly process external user data, such as unstructured data from Object Storage Service (OSS). You are no longer required to first import data into MaxCompute tables.

You can execute a DDL statement to create an external table in MaxCompute and associate the table with external data sources. This table can then act as an interface between MaxCompute and external data sources. External tables can be accessed in the same way as standard MaxCompute tables. You can fully use the computing capabilities of MaxCompute SQL to process external data.

MaxCompute allows you to create external tables to process data from the following data sources:

- Internal data sources: OSS, Tablestore, AnalyticDB, ApsaraDB RDS, Alibaba Cloud HDFS, and TDDL
- External data sources: open source HDFS, MongoDB, and HBase

21.3.6. Unstructured data processing in MaxCompute

MaxCompute has the following problems when processing unstructured data: MaxCompute stores data as volumes and must export generated unstructured data to an external system for processing.

To alleviate these problems, MaxCompute uses external tables to enable connections between MaxCompute and various data types. MaxCompute uses external tables to read and write data volumes as well as process unstructured data from external sources such as OSS.

21.3.7. Enhanced features

21.3.7.1. Spark on MaxCompute

21.3.7.1.1. Open-source platform - Cupid

21.3.7.1.1.1. Overview

MaxCompute is a big data solution independently developed by Alibaba Cloud that leads the industry in scale and stability. The big data open-source communities are actively developing big data solutions. All kinds of systems are rapidly emerging and growing to meet various requirements. To better serve users and to diversify the MaxCompute ecosystem, the MaxCompute team has developed the Cupid platform to connect MaxCompute with open-source communities. The Cupid platform integrates the diversity of open-source communities with the scale and stability of the Apsara system.

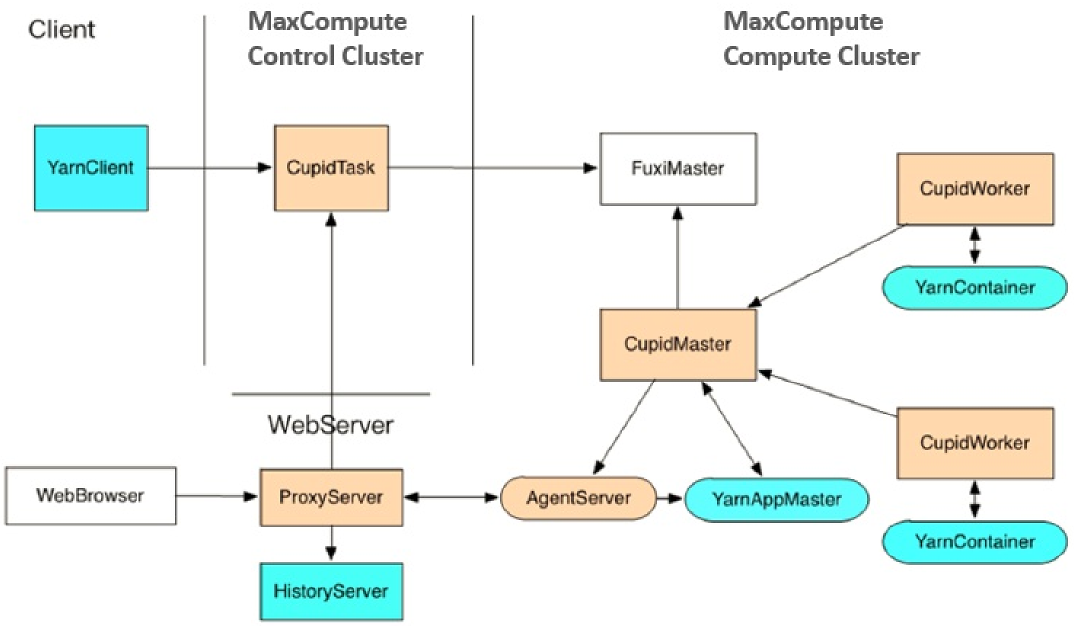
The software stacks of open-source communities and the Apsara system are similar with slight differences.

Most open-source communities use HDFS as a distributed file system, while the Apsara system uses Apsara Distributed File System. Most open-source communities use YARN as a distributed scheduling system, while the Apsara system uses Job Scheduler. On top of Job Scheduler are the computing engines designed for all kinds of scenarios. Cupid provides compatibility with open-source communities for open-source applications (such as Spark) to run on MaxCompute.

21.3.7.1.1.2. Compatibility with YARN

YARN has three application-oriented APIs: YarnClient, AMRMClient, and NMClient. YarnClient is used to submit applications to a cluster. AMRMClient is used by AppMaster to send messages to Resource Manager to request and release resources. NMClient is used to start and stop application containers.

YARN on MaxCompute



The preceding figure shows the process of submitting a YARN application to be run on MaxCompute. The yellow boxes indicate Cupid components, while the light blue boxes indicate open-source components. The procedure is as follows:

1. Use a Spark client that encapsulates the YarnClient class to access the MaxCompute control cluster and submit a job to FuxiMaster.
2. FuxiMaster starts a CupidMaster. Then, the CupidMaster starts YarnAppMaster based on the YARN protocol.
3. YarnAppMaster interacts with FuxiMaster through CupidMaster to request and release resources.
4. To start a new container, you must first use Tubo in Job Scheduler to start a CupidWorker. The CupidWorker will then start the container based on the YARN protocol.

Note Typically, YarnAppMaster provides a UI. The UI is implemented through Cupid based on a proxy mechanism.

21.3.7.1.1.3. Compatibility with FileSystem

Most open-source communities use HDFS as a distributed storage solution. The FileSystem API provided by Hadoop is compatible with Alibaba Cloud OSS and Amazon S3. Apsara Distributed File System is compatible with FileSystem API. Open-source jobs submitted to MaxCompute can be run natively on Apsara Distributed File System.

Note Apsara Distributed File System does not directly provide external services. The data in Apsara Distributed File System can only be used as intermediate job data, such as checkpoints. You can use OSS to make the data stored in Apsara Distributed File System accessible to other environments.

21.3.7.1.1.4. DiskDrive

Most open-source applications use local file systems for data processing, such as the shuffle and storage modules in Spark. In environments with large clusters, disks are important system resources. Disks must be centrally managed to ensure high availability, performance, and security. In the Apsara system, disks are centrally managed by Apsara Distributed File System. To provide local file system APIs based on Apsara Distributed File System, the Cupid team has designed and implemented the DiskDriverService system by integrating Web-based storage into MaxCompute.

21.3.7.1.2. Feature extensions

21.3.7.1.2.1. Overview

MaxCompute provides the Cupid framework to support open-source applications. This allows Spark to be run on MaxCompute. For ease of use and better integration with MaxCompute, there are several extensions available for Spark on MaxCompute to add features such as the secure isolation of open-source Spark applications, mutual access between MaxCompute data and Spark data, and support for interactions in multi-tenant clusters.

The following sections describe these extensions.

21.3.7.1.2.2. Security isolation

Spark jobs submitted to the MaxCompute computing cluster are run in sandboxes, preventing attacks on the system. A parent-child process architecture is used for the entire system. The Cupid framework runs in the parent process, and Spark runs in the child processes. When Spark requires access to system services, the parent process accesses the services on behalf of Spark by communicating with the child processes.

21.3.7.1.2.3. Data interconnection

An advantage of running Spark on MaxCompute is that resources used by Spark and MaxCompute jobs are shared across all clusters. This allows jobs to directly access their data without having to pull data across different clusters.

This data includes metadata and storage data. For security reasons, Spark must be authenticated through the MaxCompute account system before it can store MaxCompute data. **Spark on MaxCompute** provides OdpsRDD and OdpsDataFrame so that users can use Spark APIs on MaxCompute. Spark SQL has direct access to MaxCompute metadata for SQL optimization and can directly store and retrieve MaxCompute data at the physical layer.

21.3.7.1.2.4. Client mode

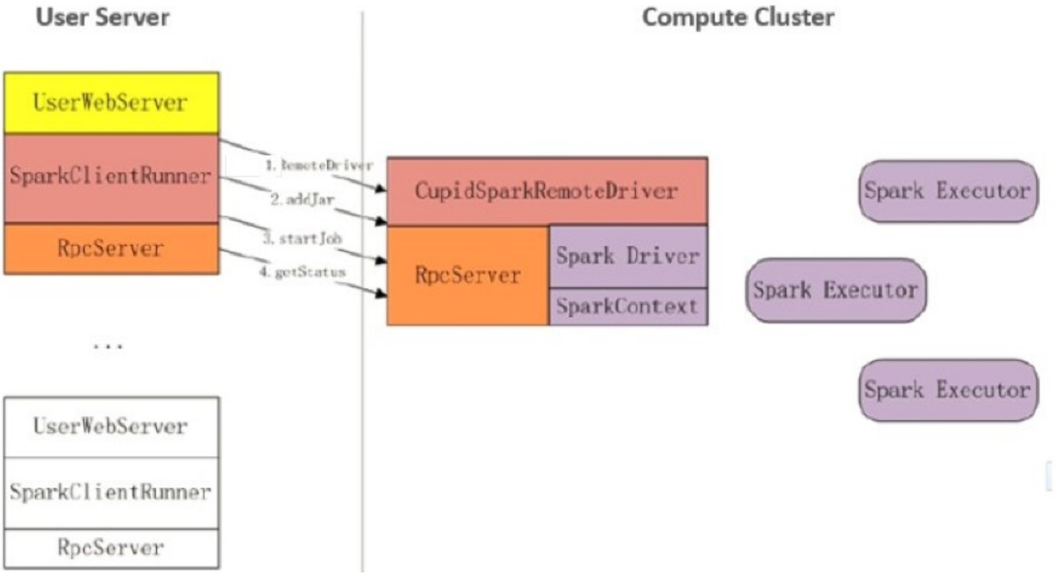
The yarn-cluster and yarn-client modes are commonly used in open-source communities for Spark-related development efforts. In yarn-cluster mode, you can submit a Spark job to a YARN cluster. After the job is run, the client generates a log that indicates the job status. In this mode, you cannot submit a job to a Spark session multiple times in real time, and the client cannot obtain the running status and result of each job. The yarn-client mode takes effect for interactive scenarios. However, to use the yarn-client mode, you need to launch the Spark driver process from the client side. You cannot use a Spark session as a service in this mode. The MaxCompute team has developed the Client mode based on **Spark on MaxCompute** to solve the preceding problems. The Client mode has the following features:

1. The client is a lightweight process that does not require you to launch the Spark driver process.
2. The client provides a set of APIs that can be used to submit jobs in real time to the same Spark session in MaxCompute clusters. It can also monitor the statuses of all jobs in the Spark session.

3. The client can build dependencies between jobs by monitoring job statuses and results.
4. You can compile an application JAR package in real time and submit it to the original Spark session through the client.
5. The client can be integrated into the Web servers of a service, and can also be scaled horizontally.

In Client mode, you need to use CupidSparkClientRunner to start a Spark session in a MaxCompute cluster. Then, you can use CupidSparkClientRunner to perform operations on the client side, such as submitting jobs and viewing the running statuses and results of the jobs. Cached data can be shared between jobs. You can also construct multiple CupidSparkClientRunner objects to interact with the same Spark session. The following figure shows the block diagram of the Spark Client mode.

Spark Client mode



The procedure for using the Spark Client mode is as follows:

1. You submit a job to a MaxCompute cluster to launch CupidSparkRemoteDriver and obtain the SparkClientRunner object.
2. You use SparkClientRunner to add the JAR package that will execute the job to RemoteDrive.
3. SparkClientRunner uses the job classname to submit the job to RemoteDriver. RemoteDriver then runs the job.
4. SparkClientRunner monitors the job status based on the job ID returned after the job is submitted.

21.3.7.1.2.5. Spark ecosystem support

The Spark ecosystem covers diverse components, including MLlib, Streaming, PySpark, SparkR, GraphX, and SQL. **Spark on MaxCompute** provides a complete Spark ecosystem that supports the scaling of original resources in open-source communities. The ecosystem provides consistent user experience with that of open-source communities. **Spark on MaxCompute** also supports access to the Spark UI and historical log files.

21.3.7.2. Elasticsearch on MaxCompute

21.3.7.2.1. Terms

term

An exact value that can be indexed. You can use a term query to search for an exact match.

text

A piece of unstructured data. Typically, a text is parsed into individual terms that are stored in an Elasticsearch index library.

cluster

A collection of one or more nodes that provide external indexing and search services. Elasticsearch is deployed in the Apsara cluster of MaxCompute. Elasticsearch clusters are a part of the Apsara cluster.

node

A logical service in an Elasticsearch cluster. A node can store data and participate in the cluster's indexing and search capabilities.

shard

A single Lucene instance which is a relatively low-level feature managed by Elasticsearch. An Elasticsearch cluster automatically manages all the shards in a cluster. When a node fails, Elasticsearch moves the shards to a different node or adds a new node.

replica

A distinct copy in Elasticsearch. Elasticsearch on MaxCompute allows you to have multiple replicas across different nodes to improve system-level availability. We recommend that you set the default number of replicas for this service to 1.

index

A collection of documents that have similar characteristics. For example, you can have an index for customer data, an index for a product catalog, and another index for order data. An index is identified by a name (that must be all lowercase) that is used to refer to the index when you perform indexing, search, update, and delete operations on the documents in the index. You can define as many indexes as you want in a single Elasticsearch cluster.

type

A logical partition of an index. You can define one or more types in an index. Typically, a type is defined as a document that has a common set of fields.

mapping

A process that defines document fields and their types as well as other index-wide settings. A mapping is similar to a schema definition in a relational database. Each index has a mapping. A mapping can either be defined in advance or automatically generated when you store a document for the first time.

document

A JSON-formatted string which is stored in Elasticsearch, similar to a row in a relational database. Each document has a type and an ID. A document is a JSON object which contains zero or more fields, or key-value pairs.

field

A simple value or a nested structure. Fields are similar to columns in relational database tables. Each field has a field type.

21.3.7.2.2. How Elasticsearch on MaxCompute works

21.3.7.2.2.1. Overview

Elasticsearch on MaxCompute is based on the open source Elasticsearch. It can run the Elasticsearch service on MaxCompute clusters.

On the MaxCompute client, you can start and manage your Elasticsearch service as needed and configure the number of nodes, disk space, memory size, and custom settings. The resources consumed by the Elasticsearch service are counted against your MaxCompute quota.

The following sections describe how **Elasticsearch on MaxCompute** functions work.

21.3.7.2.2.2. How distributed architecture works

Basic principles

An Elasticsearch cluster consists of multiple nodes. MaxCompute ensures high availability by controlling the start and stop of Elasticsearch services and nodes, allocating computing resources, and implementing failover based on a centralized scheduling mechanism.

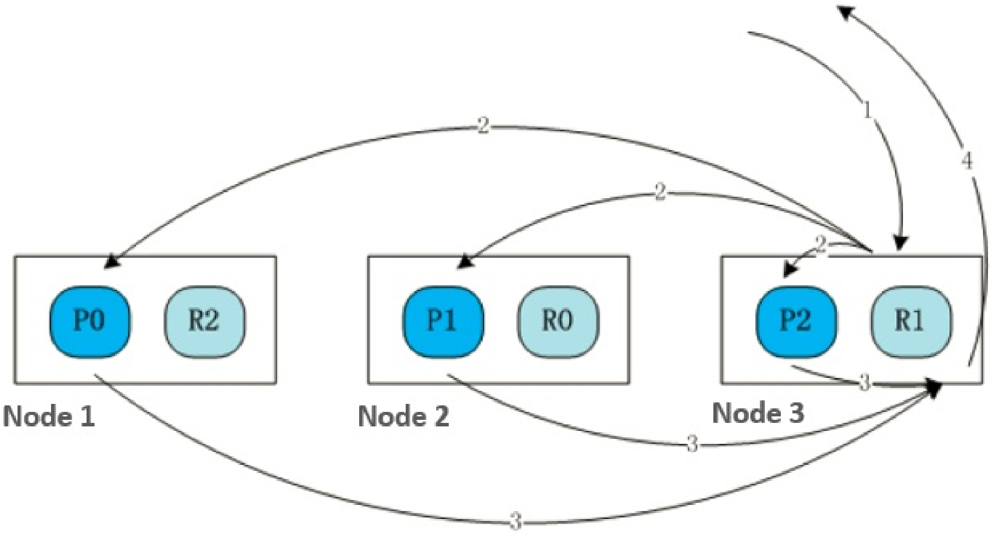
Data is replicated into multiple copies and stored in Apsara Distributed File System. This guarantees that no data is lost due to the failure of a few nodes.

An index is split into multiple shards, which are evenly distributed across multiple nodes in a cluster. The system simultaneously retrieves data shards in multiple nodes, improving retrieval performance.

Implementation process

The following figure shows the distributed retrieval workflow.

Distributed retrieval workflow



As shown in the preceding figure, each cluster consists of three nodes. The index has three shards: P0, P1, and P2. These shards are distributed across the three nodes. Each shard is replicated in 1:1 mode, generating three replicas: R0, R1, and R2. The retrieval process is as follows:

1. A user sends a retrieval request to Node 3.
2. After receiving the request, Node 3 sends a retrieval request (2) to P0, P1, and P2 based on the recorded index shard information.
3. The nodes where P0, P1, and P2 are located search for the requested information in the specified shards. A retrieval result message (3) is sent to Node 3.
4. Node 3 collects the retrieval results from other nodes and returns the retrieval results to the user in an acknowledgment message (4).

When multiple nodes are performing data retrieval at the same time, the retrieval speed is improved. The performance of distributed retrieval increases with the number of nodes.

21.3.7.2.2.3. How full-text retrieval works

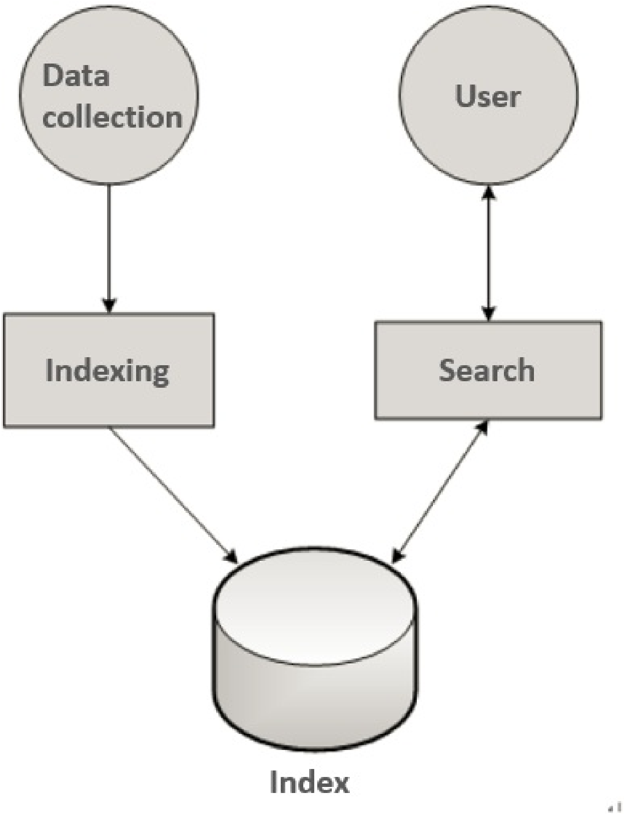
Basic principles

Full-text retrieval refers to techniques used to search for data records containing specified contents from large volumes of texts. In the retrieval process, data in texts is segmented by words, and an inverted index is created based on mappings from words to documents to allow fast document retrieval.

Implementation process

The following figure shows the full-text retrieval process.

Full-text retrieval process



The retrieval process is as follows:

1. The data collection module collects structured and unstructured data, converts the data into the field + value format, and submits the data to the indexing module.
2. The indexing module segments the data, creates inverted indexes based on a predefined indexing method, and saves the indexes. The field type, indexing method, and segmentation rules are configured on the retrieval management page.
3. The search module receives and processes user requests. Requests are parsed to obtain indexes, fields, and query statements, and then matched to records in the inverted indexes.
4. The indexing module returns data that meets user-defined requirements such as sorting rules and request quantity.

21.3.7.2.2.4. How authentication control works

Basic principles

Authentication control is implemented at the entrance used for external services to check whether users have been authorized to access the index libraries.

Implementation process

The authentication control process is as follows:

1. **Elasticsearch on MaxCompute** provides retrieval management and O&M platforms that are only accessible after logon. User account information is verified and authenticated by a centralized authentication module before logon. Any user who fails the authentication is denied access to the platforms.

2. The administrator can use the MaxCompute client to add Elasticsearch users and configure permissions for the users.
3. The system authenticates all users who attempt to access index libraries. After passing authentication, you will be able to retrieve or perform operations on data in the libraries.

21.3.8. MaxCompute multi-region deployment

This topic describes the multi-region deployment supported by MaxCompute. Control clusters are deployed in a centralized manner and used to configure resources and manage computing tasks. Compute clusters are separately deployed in each region to create projects and distribute computing tasks.

The multi-region deployment of MaxCompute has the following features:

- A MaxCompute system can manage clusters in different regions.
- Data exchanges between clusters are implemented within MaxCompute, and data replication and synchronization between clusters are managed based on configured policies.
- Metadata is stored in a centralized manner. Therefore, the infrastructure requirements, such as the network connections of different data centers, are relatively high.
- A unified account system is used.
- The development systems for big data applications, such as DataWorks, are used for clusters in all regions.
- MaxCompute must run in multi-cluster mode to support multi-region deployment.

  ? **Note** Take note of the following conditions and limits on changes to the multi-cluster mode:

  - The network bandwidth must be sufficient to support multi-region data synchronization and link redundancy.
  - Control clusters in the central region have a high latency for basic services, such as Apsara Stack DNS and Tablestore. Therefore, we recommend that you deploy basic services in the same data center to ensure that the network latency remains within 5 ms.
  - The network latency between control clusters in the central region and compute clusters in other regions must be within 20 ms.
  - Clocks must be synchronized between clusters in different regions and between servers in the same cluster.
  - The network bandwidth must be sufficient to support data replication between clusters.
  - Apsara Stack DNS is required.
  - Servers in different clusters can communicate with each other, and the clusters have similar network infrastructure (1-Gigabit or 10-Gigabit).

- The O&M and upgrades for multi-region deployment are different from those for single-cluster deployment. Multi-region deployment requires higher on-site O&M capabilities.
- MaxCompute supports cross-region multi-cluster (sub data centers) distributed computing. It uses the global job scheduling feature of the primary data center to balance the resource usage among clusters. It schedules jobs to the most appropriate cluster based on cluster information, such as the default settings, historical analysis, data distribution, and cluster load. Then, it executes the jobs and generates query results. MaxCompute supports history- and cost-based optimization policies of SQL queries. Remote clusters in unified global data management mode allow you to uniformly schedule

and manage resources that belong to multiple clusters in different data centers.

22.DataWorks

22.1. What is DataWorks?

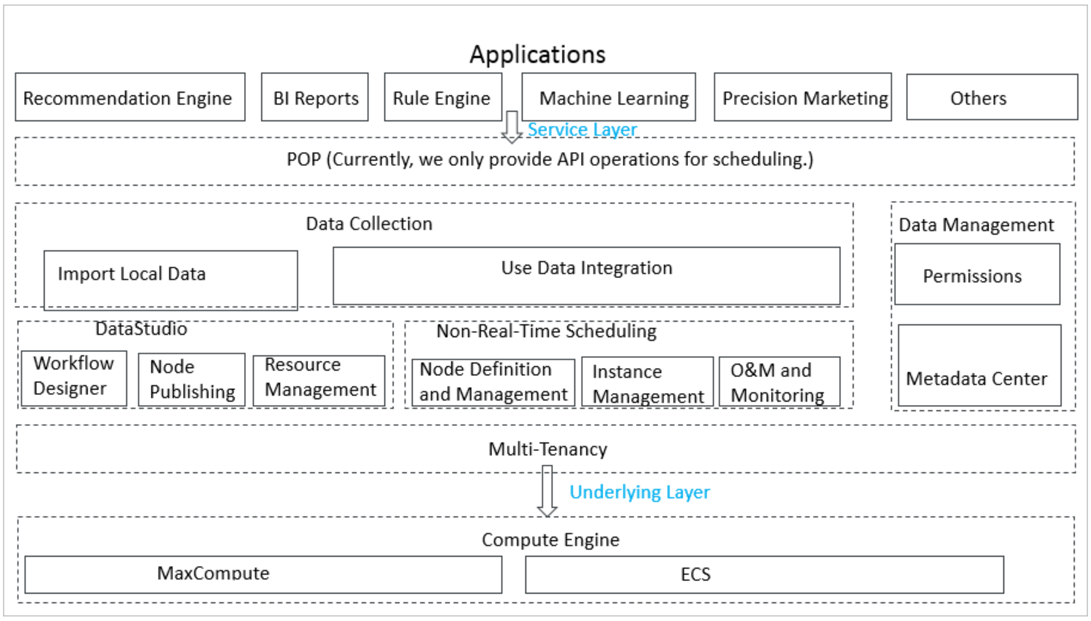
22.1.1. Overview

DataWorks is an all-in-one big data analytics and governance service released by Alibaba Cloud. It provides end-to-end solutions for enterprises and individual users to analyze, manage, schedule, govern, and apply data.

DataWorks is aimed at mining the full value of the data.

- It allows large enterprises to build petabyte-level and even exabyte-level data warehouses. The enterprises can improve their business operations by using the data integration, data asset management, and data analytics features provided by DataWorks.
- Small- and medium-sized enterprises and individual users can build data-based applications, which drive data service innovations.

Service components



DataWorks consists of an integrated development environment (IDE), a scheduling system, a data integration tool, and a data management system.

- IDE: a development tool that can be used to write SQL, MapReduce, or shell code. IDE supports collaborative development and version control. By using the visual process design tool, you can define the dependencies among different nodes.
- Scheduling system: a system that can schedule millions of batch sync nodes in a day. You can manage your nodes online, and view the logs, scheduling status, and monitoring alerts.
- Data integration tool: an integration tool that can be used to configure sync nodes between heterogeneous data stores. More than 80% of databases and storage systems provided by Alibaba Cloud and common data stores such as relational databases, FTP, and Hadoop Distributed File System (HDFS) can be configured as a source or a destination of the sync node. You can also create a node that runs periodically to synchronize data on a periodic basis.

- Data management system: a system that can be used to manage data in MaxCompute and E-MapReduce compute engines. You can manage permissions, view the data lineage, and view the metadata.

22.1.2. Scenarios

DataWorks can be applied to the construction of large data warehouses and data-driven operations.

Construction of large data warehouses

Enterprises can use DataWorks in Apsara Stack to build large data warehouses.

DataWorks can integrate petabytes of data for enterprise customers.

- Storage: provides a scalable data warehouse for petabytes and exabytes of data.
- Data integration: supports data synchronization and integration across heterogeneous data stores to eliminate data silos.
- Data analytics: supports MaxCompute-based big data processing capabilities, programming frameworks such as SQL and MapReduce, and a visualized workflow designer.
- Data management: supports unified metadata management and permission-based data access control.
- Batch scheduling: provides the scheduling of recurring nodes at different intervals, and supports scheduling of millions of concurrent nodes, error alerts, and real-time monitoring of running node instances.

Data-driven operations

- Innovative businesses: Data mining, data modeling, and real-time decision making can be implemented based on big data analytics results provided by DataWorks.
- Small- and medium-sized enterprises: DataWorks allows you to analyze data and put it to commercial use, which help enterprises generate marketing strategies.

22.2. Benefits

This topic describes the benefits of DataWorks.

Capability of processing large volumes of data

DataWorks uses MaxCompute as its computing engine, which supports a maximum of 5,000 servers in a single cluster. DataWorks can access data from different clusters, which allows you to process large volumes of data. The offline scheduling system can run millions of concurrent nodes. You can also configure rules and alerts to monitor the running of nodes in real time.

Core capabilities:

- Supports join operations for trillions of data records, millions of concurrent nodes, and petabytes (PB) of I/O throughput per day.
- Allows you to share data across clusters and data centers, and scale out clusters to a maximum of tens of thousands.
- Provides efficient and easy-to-use SQL and MapReduce engines, and supports most standard SQL syntax.
- Protects user data from loss, breach, or theft by using multi-layer data storage and access security mechanisms of MaxCompute, including triplicate backups, read/write request authentication,

application sandboxes, and system sandboxes.

Integrated data processing environment

DataWorks integrates development, scheduling, O&M, monitoring, and alerting for nodes, and management of data.

Core capabilities:

- Provides you with all the required features for data processing.
- Provides a visual designer similar to Kettle for you to design and edit workflows.
- Provides a collaborative development environment. You can create and assign roles for varying nodes, such as development, online scheduling, O&M, and data permission management, without locally processing data and nodes.

Integration from heterogeneous data sources

DataWorks can read data from 11 heterogeneous data sources and write data to 12 heterogeneous data sources. You can also configure dirty data filtering and bandwidth throttling.

Core capabilities:

- Supports data reading from data sources of the following types: MySQL, Oracle, SQL Server, PostgreSQL, ApsaraDB RDS, PolarDB-X, MaxCompute, FTP, Object Storage Service (OSS), Hadoop Distributed File System (HDFS), Dameng, and Sybase.
- Supports data writing to data sources of the following types: MySQL, Oracle, SQL Server, PostgreSQL, ApsaraDB RDS, PolarDB-X, MaxCompute, AnalyticDB, Memcache, OSS, HDFS, Dameng, and Sybase.
- Supports dirty data filtering and bandwidth throttling.
- Supports node recurring.

Web-based software

DataWorks is out-of-the-box. You can use it whenever an internal network or the Internet is available.

Multitenancy

DataWorks uses multitenancy to isolate data among tenants. Each tenant separately manages their own permissions, data, resources, and members.

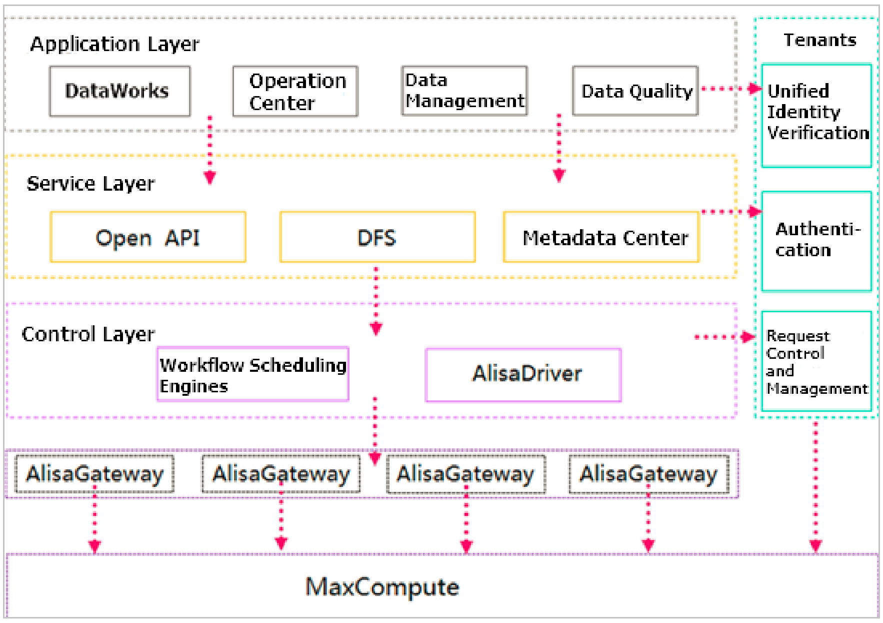
Open platform

DataWorks provides all modules as components and services. You can use DataWorks APIs to develop extra features for DataWorks.

22.3. Architecture

This topic describes the system architecture, security architecture, and multi-tenancy model of DataWorks.

System architecture



DataWorks adopts the design of components and services, and consists of the following three layers:

- Control layer: the core of batch data processing in DataWorks. The workflow scheduling engine generates and runs node instances. AlisaDriver coordinates and controls the running of all nodes.
- Service layer: provides services for the application layer and other external applications.
- Application layer: runs on top of the service layer, and provides the graphical interface for user interactions.

Security architecture

The security architecture of DataWorks features error proofing, basic security, and optional security tools.

- Error proofing ensures proper running of DataWorks during coding, deployment, and configuration.
- Basic security ensures the security of data for DataWorks by using features such as resource isolation among tenants, user identity verification, authentication, and log auditing.
- Optional security tools in DataWorks allow you to customize security policies for the protection and management of your system and data.

Multi-tenancy

DataWorks adopts a multi-tenancy model.

- Storage and computing resources are scalable. You can manage your own resources and request resource quotas as needed.
- Tenants are isolated. Each tenant separately manages its own data, permissions, accounts, and roles.

22.4. Services

22.4.1. DataStudio

DataWorks DataStudio provides an all-in-one IDE. In DataStudio, you can build data warehouse models, query data, develop the extract-transform-load (ETL) process, and develop algorithms. In addition, it supports collaborative development and file version control.

Features

- Provides a visual workflow designer similar to Kettle for you to design workflows and manage nodes in each workflow.
- Allows you to upload local files.
- Supports data integration from heterogeneous data sources.

> ⓘ Note
>
> Data synchronization nodes support the following data sources:
>
> - Synchronization nodes can read data from the following data sources: MySQL, Oracle, SQL Server, PostgreSQL, ApsaraDB RDS, PolarDB-X, MaxCompute, FTP, Object Storage Service (OSS), Hadoop Distributed File System (HDFS), Dameng, and Sybase.
> - Synchronization nodes can write data to the following data sources: MySQL, Oracle, SQL Server, PostgreSQL, ApsaraDB RDS, PolarDB-X, MaxCompute, AnalyticDB, Memcache, OSS, HDFS, Dameng, and Sybase.

- Provides a web-based programming and debugging environment that allows you to create SQL, MapReduce, shell (limited support), and synchronization nodes.
- Supports node deployment across MaxCompute projects. You can deploy nodes and code to the scheduling system across different workspaces.
- Adopts version control, node locking, and conflict detection mechanisms to facilitate collaborative development.
- Allows you to search for and use MaxCompute tables, resources, and user-defined functions (UDFs).

22.4.2. Data Map

Developed based on Data Management, Data Map uses roles to control the permissions for using different features, such as the permissions for creating and previewing data. Data Map helps you build a better enterprise-level knowledge base.

Data Map allows you to query tables, view details of tables, and manage permissions on tables. You can also add tables to your favorites. For information about Data Map, see Data Governance > Data Map in *DataWorks User Guide*.

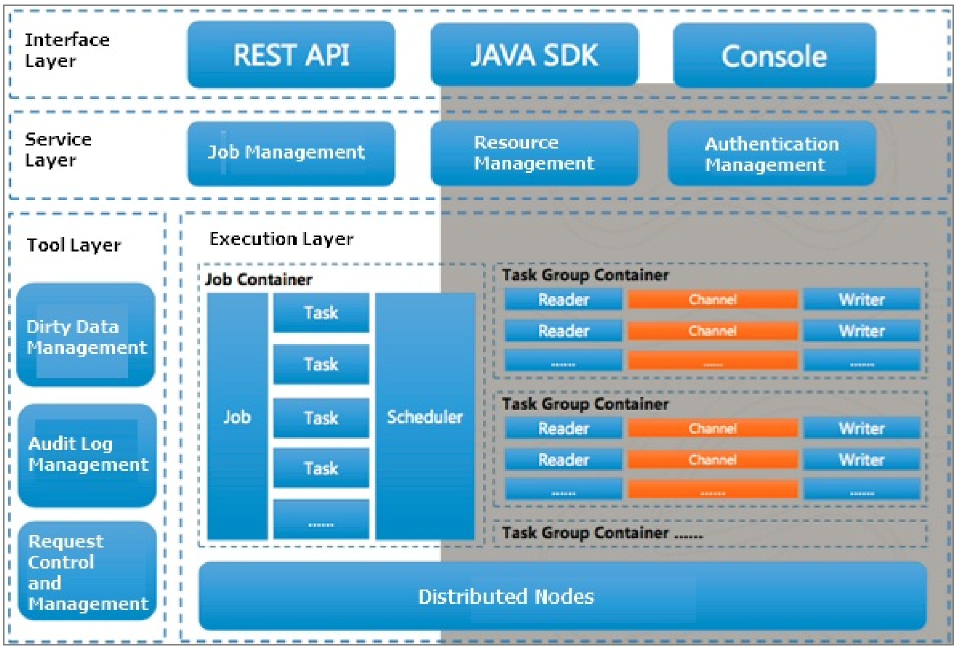
22.4.3. Data Integration

Data Integration is a data synchronization platform that provides stable, efficient, and scalable services. It provides transmission channels for batch data stored in MaxCompute, AnalyticDB, and Realtime Compute. Data Integration implements fast integration on data from heterogeneous data stores.

Data Integration adopts the framework and plug-in model. The framework is used for common operations in data synchronization and transmission. The plug-ins are used to read and write data. Data Integration supports the following plug-ins:

- Reader: reads data from data stores.
- Writer: writes data to data stores.

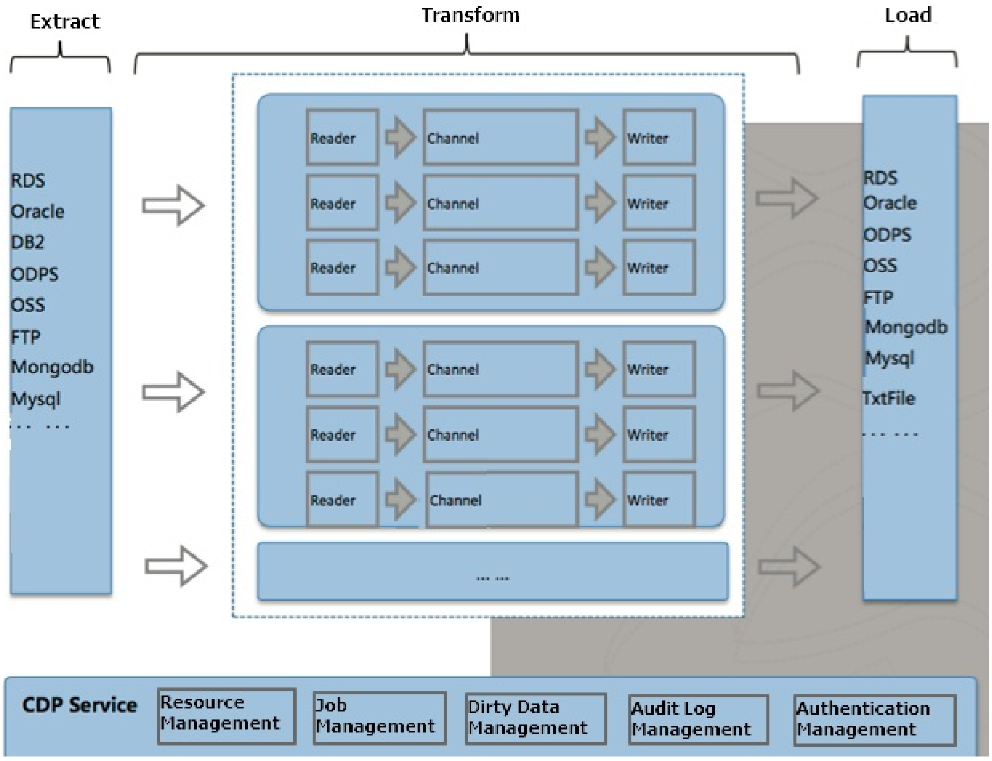
You can develop readers and writers for Data Integration to support more data stores.



Data Integration consists of the interface layer, service layer, and tool and execution layer.

- The interface layer provides three methods of using the Data Integration service: RESTful API, Java SDK, and console.
  - The RESTful API method can be used in multiple language environments. If you are a Java developer, we recommend that you use the Java SDK method to avoid manual processing of authentication, authorization, and underlying HTTP calls.
  - The console is developed based on the command-line tool, which allows you to use the majority of Data Integration functionalities.
  - Data Integration provides a web interface that is developed based on the RESTful API, which is recommended for developers.
- The service layer includes resource management, node management, and authentication management. For more information, see the service overview.
- The tool and execution layer is the core of Data Integration. This layer implements the ETL process. All sync nodes that are committed to Data Integration are run on the execution layer. The execution layer uses DataX as the synchronization engine.

  ETL process



Features

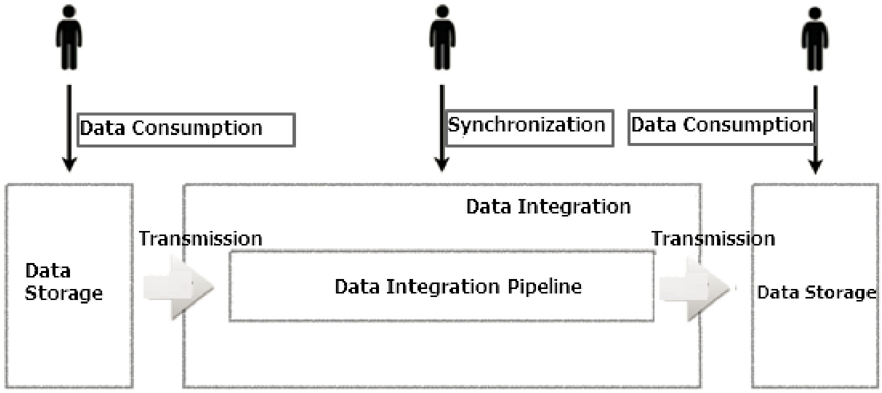
- Various types of data stores
  - Relational databases: MySQL, SQL Server, PostgreSQL, DRDS, Oracle, and general relational databases
  - NoSQL databases: Tablestore and Memcache
  - Big data storage systems: MaxCompute and AnalyticDB for MySQL
  - Semi-structured storage systems: OSS, HDFS, and FTP

  You can use the following Java Database Connectivity (JDBC) URLs when you configure connections to general relational databases such as Dameng, Db2, and PPAS:

  - Dameng: jdbc:dm://ip:port/database
  - Db2: jdbc:db2://ip:port/database
  - PPAS: jdbc:edb://ip:port/database

  Data Integration supports periodic batch synchronization. For example, you can configure a sync node that runs on a daily, weekly, or monthly basis. When the batch sync node starts, a snapshot of source data is taken. The system then reads data from the snapshot and writes the data to the destination data store. Each batch sync node has a lifecycle.

Data Integration processes only data synchronization and transmission. The complete transmission process is under the control of the Data Integration synchronization cluster model. The channels and data flows involved in the synchronization processes are isolated from users. Data Integration does not provide an API for data analysis. To perform data analysis, use DataStudio.



- Consistent data quality
    - Supports conversions between different data types.
    - Accurately identifies, filters, collects, and displays dirty data to ensure the quality of data.
    - Supports node performance reporting, which helps you track node status, such as data volume and dirty data.
- Efficient data transmission
    - Supports one-way data channels, and allows a single process to reach the maximum data transfer rate up to 200 Mbit/s on each server.
    - Adopts a distributed architecture and supports transmission for gigabytes to terabytes of data.
- User-friendly control experience
    - Implements accurate control of channels, record streams, and byte streams.
    - Allows you to rerun any threads, processes, and tasks that fail.
- Clear core design
    - Provides a professional framework and an efficient execution engine. The engine supports common plug-ins, standardizes the process of developing plug-ins, and automatically detects new plug-ins.
    - Provides clearly defined and easy-to-use plug-ins that allow developers to focus on the business instead of the framework.

22.4.4. Tenant management

You can manage workspaces, members, and permissions.

- Workspace configuration

  The Project Management page displays basic workspace settings.

    - Sandbox whitelist: Configure the IP addresses and domains that can access the workspace.
    - Compute engine: View the information about existing compute engines.
- Member management

On the Members page, you can assign or revoke a role from specified members.

- Permission management

  On the Permissions page, you can view the system permissions and their categories.

22.4.5. Data Quality

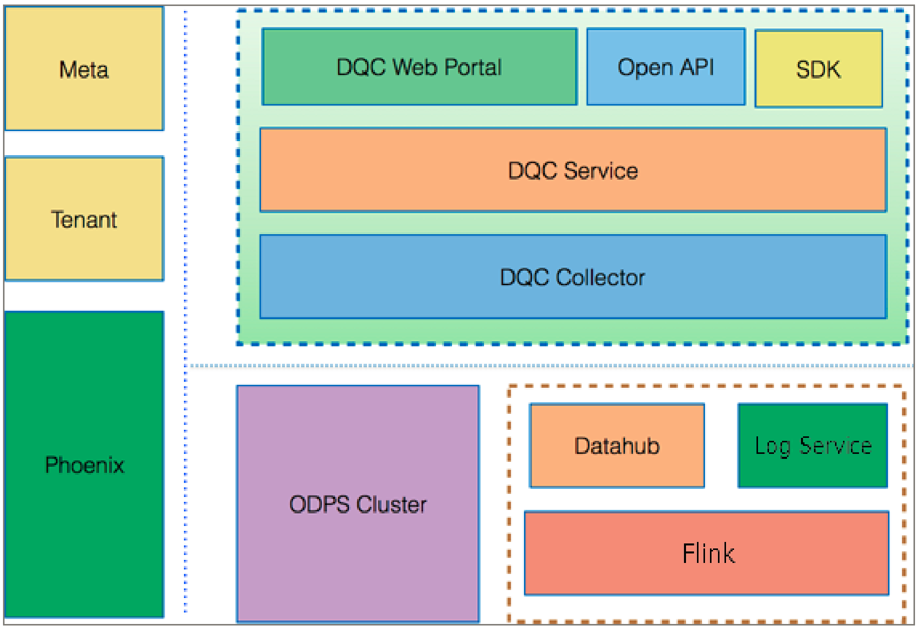
22.4.5.1. Overview

Data Quality is a platform that provides data quality check and management services. You can use it to monitor both real-time and batch data during the entire data processing cycle.

When you use Data Quality to monitor real-time data, it can detect discontinuity, delay, and other user-defined data issues in data streams. When you use Data Quality to monitor batch data, it can detect abnormal data in the production process, protect downstream data from being affected by abnormal data, and promptly notify you about the abnormal data. This helps ensure the correctness of your data.

Data Quality requires the access to the metadata, fields, and tables, and requires user and tenant management. In the scenario of monitoring batch data, Data Quality uses MaxCompute as the compute engine. In the scenario of monitoring real-time data, Data Quality uses the Flink framework as the streaming data processing tool. Data Quality consists of three components: the web portal, the check service, and the data collection service.

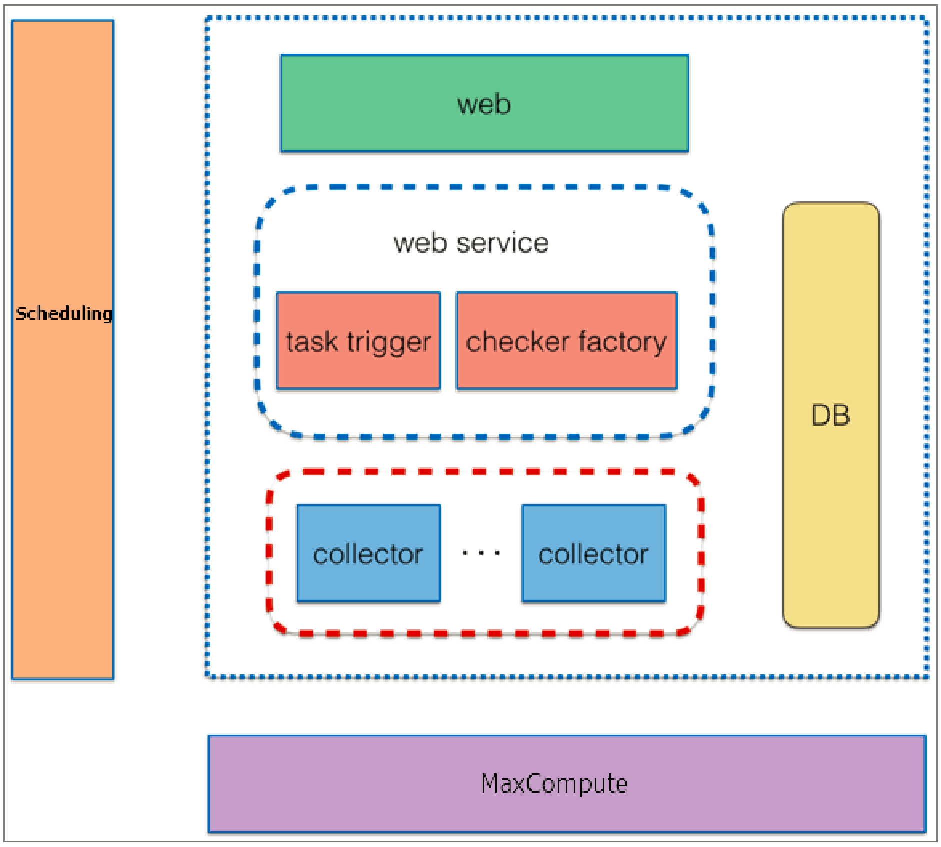
Data Quality architecture



22.4.5.2. Use Data Quality to monitor batch data

This topic describes the architecture, working principles, and benefits of using Data Quality to monitor batch data.

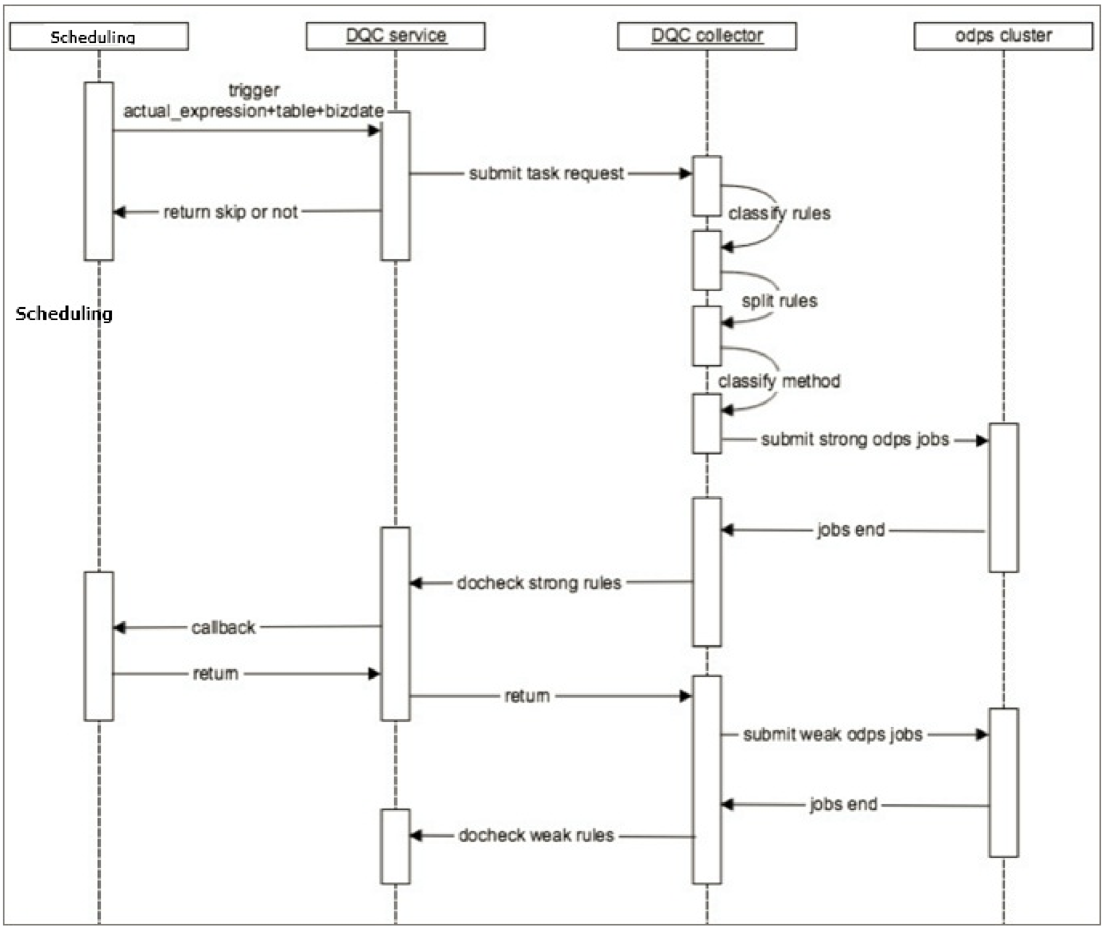
Architecture



Data Quality consists of the web UI, web service, and collector modules.

- Web UI: provides a graphical interface for user interactions. It provides features such as rule management, search by node, subscription management, dashboard, permission control, and cache management.
- Web service: provides access to databases, checks data quality, parses nodes, and triggers nodes. The checker factory module checks samples by using quality check logic such as comparison of fixed value, fluctuation, and variance detection.
- Collector: consists of multiple data collection engines that obtain data samples based on user specified rules. Data collection engines classify the rules based on potency, rule types, and sampling methods. Before the data collection engines send the rules to MaxCompute to obtain data samples, the data collection engines apply logical splitting and combination to the rules.

How it works



Data Quality monitors batch data in the following way:

1. The scheduling system sends a request that triggers the service module to check the quality of data in the specified partitions of a table. The request contains the partition expression, table information, and node schedule.
2. Based on the partition expression, a server in the service module obtains the set of rules that are applied to the current node. The server submits a request for obtaining data samples to data collection engines and returns the request result to the scheduling system. The scheduling system first allocates resources to run nodes that are associated with strong rules.
3. The data collection engines further classify the set of rules based on potency, rule types, and sampling methods. The MaxCompute cluster collects data samples based on the sampling methods.
4. After the data collection engines finish data sampling based on strong rules, the data collection engines instruct the service module to check data quality. After the quality check, the service module sends the check results to the scheduling system, and the scheduling system determines whether to block the node.
5. After the quality check by using strong rules, the service module returns the results to the data collection engines. The data collection engines continue the sampling process, and send the processed data for check based on weak rules. After the weak rule check is complete, the quality check ends.

Benefits

- Data Quality provides built-in rule templates and comprehensive data quality metrics.

The templates support filed and table level rules with a fluctuation threshold or fixed value comparison. You can create rules from the templates to check whether data entries are null or unique or use discrete values, the maximum, minimum, average, or sum to evaluate the data quality. You can also create custom rules for special requirements.

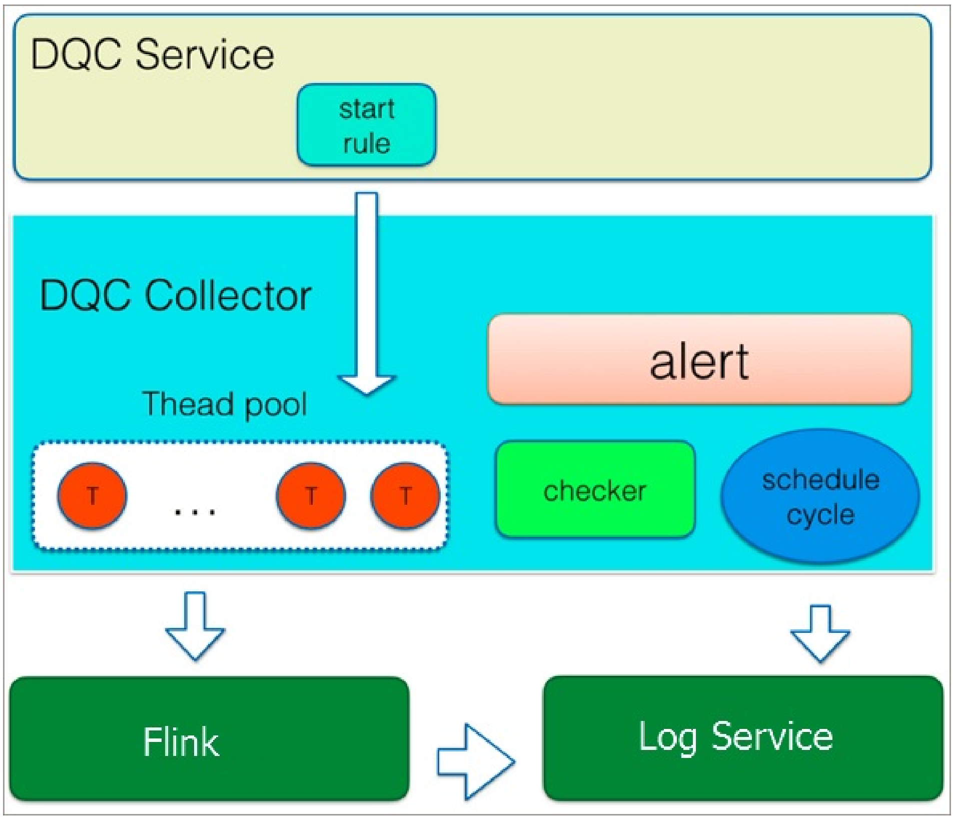
- Data Quality clusters are horizontally scalable. You can add servers if Data Quality reaches the maximum concurrency. Data Quality also includes a reliable fault-tolerance system that ensures that data collection tasks are accurate and consistent.
- Data Quality supports rule classification based on potency and severity levels.

  When you use Data Quality to monitor batch data, you can classify rules into weak and strong rules based on potency. You can also set thresholds to reflect the warning and error severity levels of check results based on the deviation from the expected value. When strong rule check results show a significant deviation from expected values, the node is blocked to protect downstream data against dirty data. This ensures the correctness of data during the data processing cycle.
- Data Quality provides a potency-based execution mechanism that first runs the nodes that are associated with strong rules. The data collection engine supports running nodes based on the potency.
  - If available resources are limited, this mechanism ensures that you first run nodes that are associated with strong rules.
  - If available resources are sufficient, this mechanism allows nodes that are associated with weak rules to run.

22.4.5.3. Use Data Quality to monitor real-time data

This topic describes the architecture, working principles, and benefits of using Data Quality to monitor real-time data.

Architecture



Rules for monitoring real-time data are converted into Flink SQL statements. Data Quality uses Flink to read data from DataHub and write check results to Log Service. The collector module of Data Quality regularly obtains abnormal data from Log Service, writes the data to Redis, and then triggers alerts. The service module of Data Quality synchronizes the alerts from Redis to other databases for users to query.

How it works

Data Quality monitors real-time data in the following way:

1. After you enable a rule, the service module creates a Logstore. The service module uses an SQL parser to declare a dimension table used for referencing a DataHub topic. The service module uses a rule converter to generate a CREATE TABLE statement and combine table operations. Then, the service module submits a Flink node and updates the next quality check time.
2. One of the servers in the service module first establishes a lock to serve as the master. The master collects data from DataHub topics on a regular basis and sends the data to the collector module for quality check.
3. The collector module uses a LogHub consumer to subscribe to the Logstore. Then, the collector module writes abnormal data to Redis, and determines whether to send alerts.
4. The service module starts the Quartz scheduler worker service, and writes the data from Redis to another database for users to query.

Benefits

- Monitors data discontinuity and latency in real time in multiple scenarios. It can join multiple streams and dimension tables from one data store, and allows you to write Flink SQL statements to define your own business rules.
- Supports monitoring on data latency at the level of seconds.

- Allows you to set thresholds at the warning and error severities. This helps you identify the deviation of check results from expected values.
- Allows you to set the minimum alert interval and the number of alerts to reduce redundant notifications.
- Provides you with more reliable alert information because raw alerts are Hash de-duplicated. This ensures the idempotence of data during the real-time computing process and avoids repeated notifications.

22.4.6. Data Asset Management

Data Asset Management allows you to view the metadata collected in Data Map. You can also modify the categories for the metadata, add business descriptions to tables, and view the metadata.

Data Asset Management provides you with an overview of your data assets. It requires that data be synchronized by using Data Integration and processed by using DataStudio before you manage your tables and API operations stored in your business system and DataWorks.

22.4.7. Real-time analysis

The real-time analysis feature allows you to query and preview data. This feature is suitable for data analysis and data exploration.

You can create, rename, and delete directories and files.

1. Click the **Run** icon to run the SQL statements.
2. View the running result.

22.4.8. DataService Studio

DataService Studio supports API hosting, authentication, authorization, and management. You can create APIs for tables and publish the APIs by using the API Gateway service.

DataService Studio provides the following features:

- Supports various data sources, including relational databases, AnalyticDB, and NoSQL databases.

  Supported data sources: MySQL, Oracle, SQL Server, PostgreSQL, ApsaraDB RDS, PolarDB-X, AnalyticDB, Tablestore, MongoDB, and Lightning.
- Provides the codeless UI, which can be used to generate APIs without writing code.
- Provides the code editor, which can be used to create APIs by compiling SQL statements.
- Provides accurate access control. You can customize permissions on APIs, table rows, and table columns.
- Allows you to call API operations by using API Gateway or HTTP requests.
- Supports a variety of network environments, such as local private networks, virtual private clouds (VPCs), and the classic network.
- Allows you to manage APIs, such as managing API groups and APIs, publishing APIs, and removing APIs.
- Supports API isolation by workspace or tenant.
- Allows you to register, manage, and present APIs.
- Supports a variety of API execution environments, including standalone environments and the EAS

container service.

- Allows you to debug APIs online. You can view API call information and the performance in real time.

22.4.9. Intelligent Monitor

Intelligent Monitor is a system that monitors and analyzes nodes in DataWorks. Intelligent Monitor sends alerts based on specified rules, times, methods, and alert contacts. It automatically selects the most appropriate alerting time, notification methods, and alert contacts.

Intelligent Monitor has the following benefits:

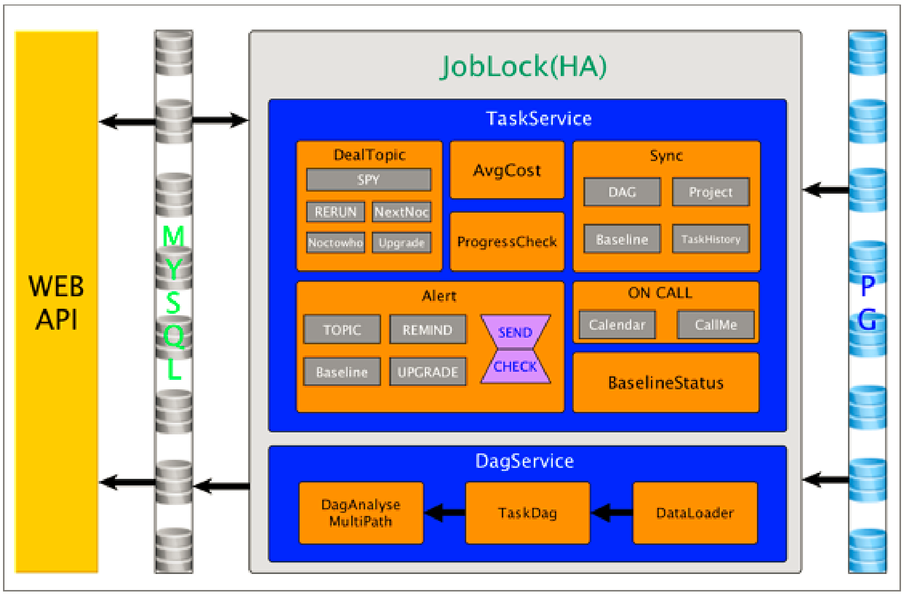
- Improves your efficiency on configuring alert triggers.
- Prevents invalid alerts from bothering you.
- Automatically covers all important nodes for you.

A conventional monitoring system allows you to configure monitoring rules, but cannot meet the requirement of DataWorks due to the following causes:

- DataWorks has numerous nodes, so it is difficult for you to find out the nodes to be monitored. Dependencies between the nodes are complex. Even if you know the most important nodes, it is difficult to find out all ancestor nodes of these nodes and monitor them all. In this case, if you monitor all nodes, a large number of invalid alerts may be generated. In consequence, you may miss those useful alerts.
- The alerting method varies with monitored nodes. For example, some monitoring tasks require the relevant nodes to run for more than 1 hour before alerts are triggered, whereas other monitoring tasks require the relevant nodes to run for more than 2 hours. It is complex to set an alerting method for each node separately, and it is difficult to predict the alert threshold value for each node.
- The alerting time varies with monitored nodes. For example, alerts for unimportant nodes can be reported after the working hours start in the morning but alerts for important nodes must be immediately reported even in off hours. It is hard for a conventional monitoring system to distinguish the importance of nodes.
- Different alerts require different operations to turn off.

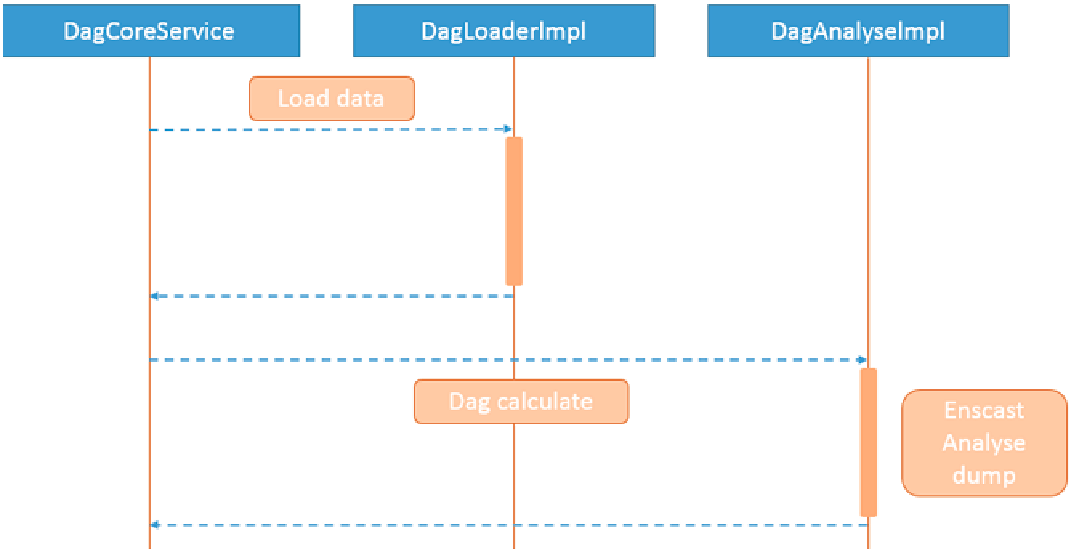
Intelligent Monitor provides comprehensive monitoring and alerting logic. You only need to provide the names of important nodes in your business. Then, Intelligent Monitor automatically monitors the entire running process of your nodes and creates standard alert triggers for them. In addition, you can customize alert triggers by completing basic settings.

Architecture



- DagService: analyzes all nodes in each directed acyclic graph (DAG) based on the baseline settings. DagService then collects information such as the estimated completion time, the key path, the required completion time, and whether to suspend a node. The information collected by DagService provides the basis for TaskService.
- TaskService: runs different nodes based on the information provided by DagService, including estimating the completion time, acquiring and fixing events, and customizing baseline alerts.
- WebService: provides the HTTP API that can be called to send requests. You can call API operations to view the Intelligent Monitor information, such as baseline instances, alert information, events, and gantt charts.

How it works



DagService collects the information of all nodes on each DAG based on the baselines and the average running time of each node. The information contains the estimated completion time, the required completion time, the key path, whether to suspend a node, and whether the node is a child of a suspended node.

TaskService runs nodes based on the node configuration and the information provided by DagService. The database lock ensures that one node is run by only one server. When a server is down, another server takes over the node, which ensures the high availability of the monitoring service.

22.4.10. Scheduling system

22.4.10.1. Overview

The scheduling system is one of the core systems in DataWorks. It is responsible for scheduling all batch sync nodes based on the specified time and the dependencies. The scheduling system provides the following features:

- Schedules millions of nodes.
- Adopts a distributed execution architecture so that the number of concurrent nodes can be linearly expanded.
- Supports different granularities for the scheduling interval, such as minute, hour, day, week, month, and year.
- Supports same-cycle dependency, cross-cycle dependency, and self-dependency between nodes.
- Supports special operations such as dry runs, node suspension, and one-off nodes.
- Allows you to create and run an ad-hoc workflow.
- Displays a workflow in a DAG, which provides you with a clear view for O&M.
- Supports real-time monitoring and alters. Alerts can be sent by text message and email.
- Supports administrative operations such as rerunning a node or multiple nodes at a time, terminating processes, setting the node status to Successful, and suspending nodes.
- Generates retroactive data for multi-cycle instances that are run in sequence.
- Provides an interface that displays the summary of global node details, including the number of scheduled nodes, the number of failed nodes, the number of running nodes, top 10 scheduled nodes by computing resource consumption, top 10 scheduled nodes by execution duration, and node distribution by type.

22.4.10.2. Terms

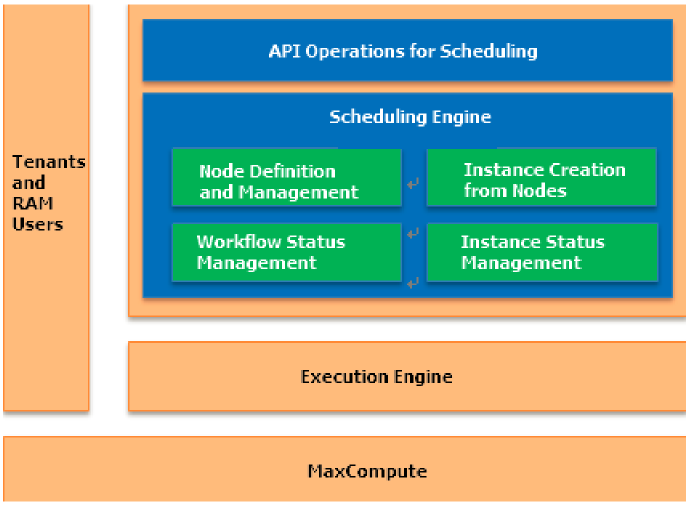
This topic describes the terms of the scheduling system.

- Node: A node represents a batch synchronization task in the scheduling system. Node properties include the node type, the code version, the specified time for running the node, and the dependencies between nodes.
- Instance: An instance is generated each time a node is run in the scheduling system to track the running of the node. An instance contains the runtime information such as the instance status and the time when the status changes.
- Workflow: A workflow is composed of several interdependent instances. The scheduling system consolidates all instances in a day into a workflow for unified management. A workflow has its own status, which is determined by the status of each instance in the workflow.

22.4.10.3. Architecture

This topic describes the architecture of the scheduling system.

The following figure shows the architecture of the scheduling system and its relationship with other systems.


Tenants and RAM Users
API Operations for Scheduling
Scheduling Engine
Node Definition and Management
Instance Creation from Nodes
Workflow Status Management
Instance Status Management
Execution Engine
MaxCompute

The scheduling engine is the core of the scheduling system. It contains four modules.

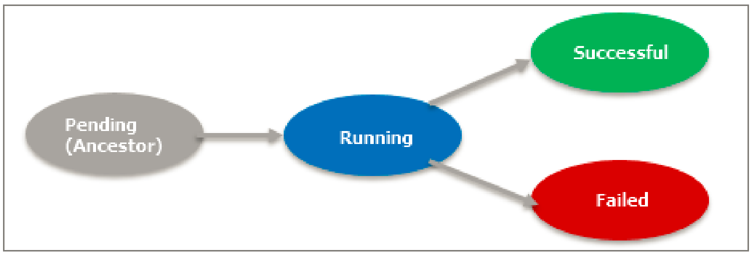
- The node definition and management module maintains node definitions submitted by users, including the code, the specified time for running the node, and the dependencies. An instance is generated from the node configurations at a fixed time every day.
- The instance state management module manages the state changes after an instance runs.
- The workflow state management module maintains the state changes after a workflow runs. A workflow is a set of instances with dependencies.
- The scheduling system allows other systems to add, delete, modify, and query its internal scheduling data by calling API operations.

The resources that are used by the scheduling system are isolated among tenants. Before a node instance runs, the scheduling system schedules the instance to the execution engine.

22.4.10.4. State machines

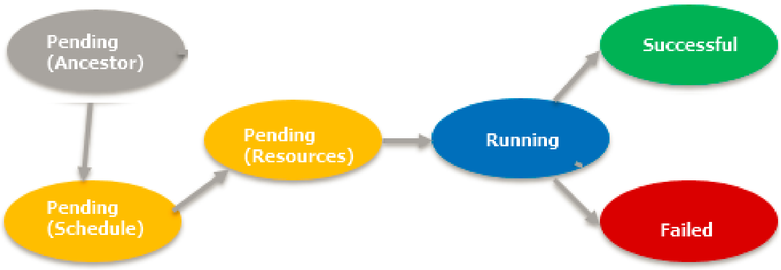
This topic describes the state machines of workflows and node instances.

Workflow state machine



- A workflow has four states: Pending (Ancestor), Running, Successful, and Failed.
- The initial state of a workflow is Pending (Ancestor). At this time, all instances in this workflow are in the Pending (Ancestor) state. When the workflow is called by the scheduling system, its state changes to Running and the root instance of the workflow runs.
- When an instance in the workflow fails, the state of the workflow changes to Failed.
- When all instances in the workflow are in the Successful state, the state of the workflow changes to Successful.

Node instance state machine



- A node instance has six states: Pending (Ancestor), Pending (Schedule), Pending (Resources), Running, Successful, and Failed.
- The initial state of a node instance is Pending (Ancestor). When it is called by the scheduling system, the system checks whether all its parent nodes are in the Successful state. If yes, the state of the instance changes to Pending (Schedule).
- The node instance is called at the time that is specified for running the node. The instance is then sent to the execution engine and its state changes to Pending (Resources).
- The execution engine allocates resources to the instance. The instance runs, and the scheduling system changes the state of the instance to Running. The execution engine sends the result to the scheduling system, and then the scheduling system changes the instance state to Successful or Failed.

22.4.10.5. Node dependencies

You can configure dependencies for nodes based on your business requirements.

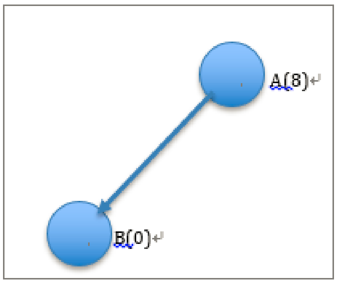
Same-cycle dependency

This is the most common scenario where an instance depends only on its parent instances in the same day. You can configure the following dependencies: A daily-run instance depends on another daily-run instance, a daily-run instance depends on an hourly-run instance, an hourly-run instance depends on a daily-run instance, or an hourly-run instance depends on another hourly-run instance.

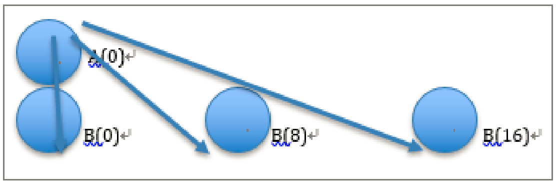
If an hourly-run instance depends on another hourly-run instance, three situations can occur: The number of parent instances is equal to the number of child instances, the number of parent instances is greater than the number of child instances, or the number of parent instances is less than number of child instances. The following examples show all the situations.

Note In the following examples, all A nodes are parent nodes, and all B nodes are child nodes.

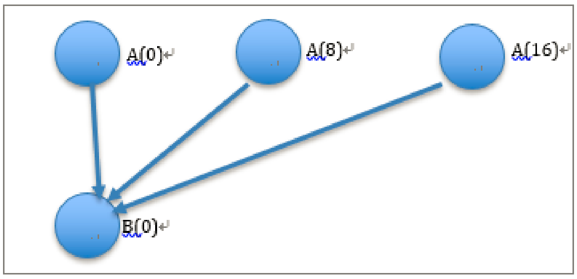
- A daily-run instance depends on a daily-run instance. The B node is specified to run at 08:00. The A node is specified to run at 00:00.



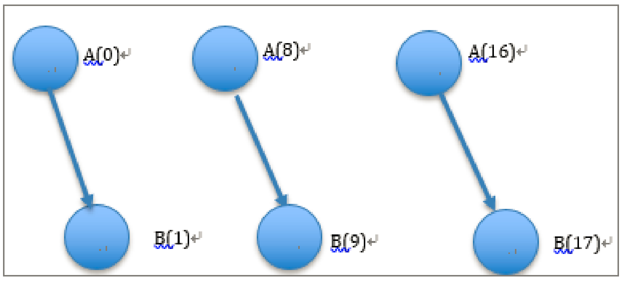
- An hourly-run instance depends on a daily-run instance. The B node is specified to run at 00:00, 08:00, and 16:00. The A node is specified to run at 00:00.



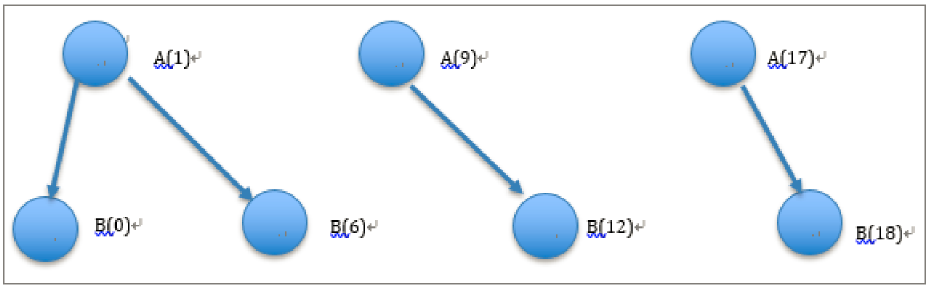
- A daily-run instance depends on an hourly-run instance. The B node is specified to run at 00:00. The A node is specified to run at 00:00, 08:00, and 16:00.



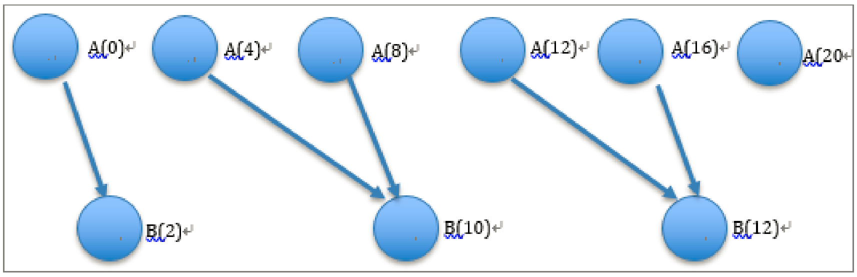
- An hourly-run instance depends on an hourly-run instance, and the number of parent instances is equal to the number of child instances. The B node is specified to run at 01:00, 09:00, and 17:00. The A node is specified to run at 00:00, 08:00, and 16:00.



- An hourly-run instance depends on an hourly-run instance, and the number of parent instances is less than the number of child instances. The B node is specified to run at 00:00, 06:00, 12:00, and 18:00. The A node is specified to run at 01:00, 09:00, and 17:00.



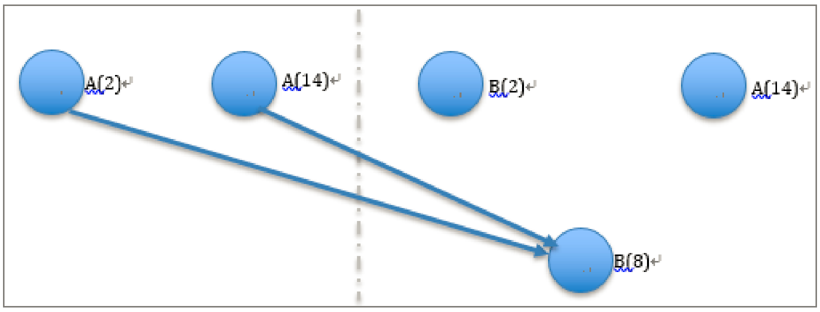
- An hourly-run instance depends on an hourly-run instance, and the number of parent instances is greater than the number of child instances. The B node is specified to run at 02:00, 10:00, and 18:00. The A node is specified to run at 00:00, 04:00, 08:00, 12:00, 16:00 and 20:00.



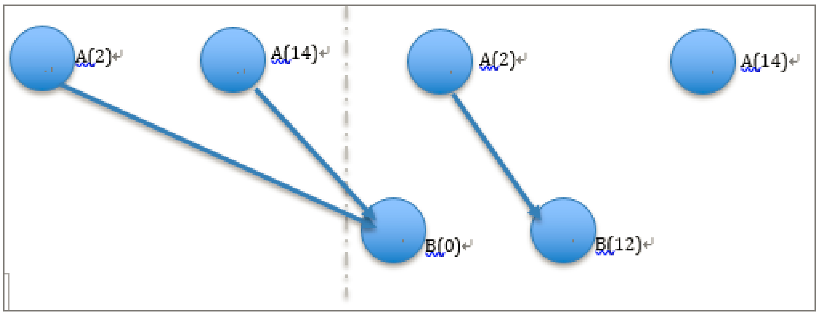
Cross-cycle dependency

You can configure cross-cycle dependency if the data processing operation requires the result of the data processing operation on the previous day.

- In most cases, you only need to configure the dependency between the current instance and the instance in the last day. Suppose that the A node is specified to run at 02:00 and 14:00, and the B node is specified to run at 08:00.

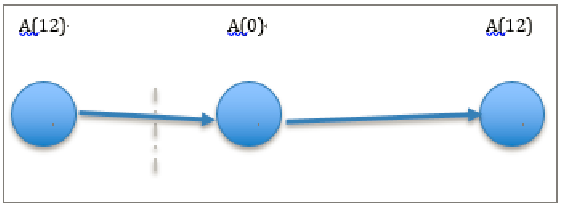


- The same-cycle dependency and the cross-cycle dependency can both exist. Suppose that the A node is specified to run at 02:00 and 14:00, and the B node is specified to run at 00:00 and 12:00.



Self-dependency

If a node instance depends on the instance that is generated from the same node in the last cycle, you must configure self-dependency. The following figure shows the dependencies in the situation where the A node is specified to run at 00:00 and 12:00.



23.Realtime Compute

23.1. What is Realtime Compute?

23.1.1. Background

Realtime Compute has its beginnings in the real-time big screen service of Alibaba Group during the Double 11 Shopping Festival. The big screen service allows you to view sales data during the shopping festival in real time on big screens. With five years of experience and development, the small team that once provided the real-time big screen service and limited real-time reporting services has become an independent and reliable cloud computing team. Realtime Compute provides an end-to-end cloud solution for stream processing based on years of experience in real-time computing products, architecture, and business scenarios. We strive to help more enterprises with real-time big data processing.

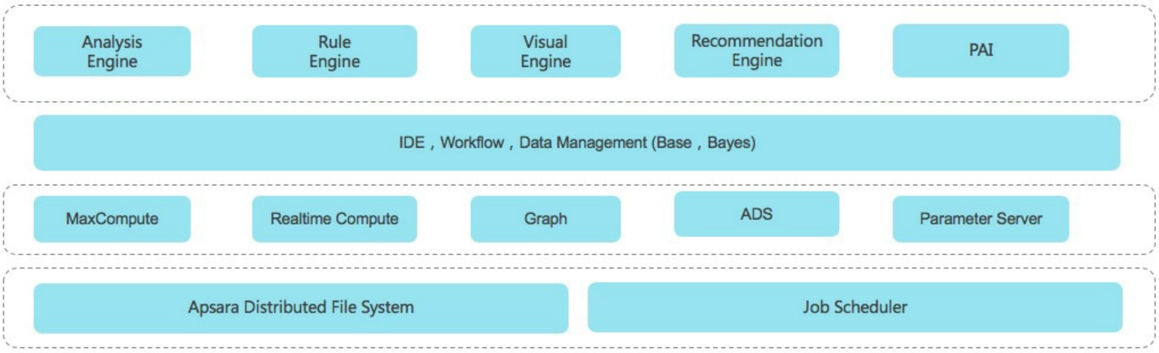
We previously used the open source Storm system to support the big screen service of Alibaba Group during the Double 11 Shopping Festival. We also developed stream processing code based on Storm. In these early stages, the stream processing service was provided on a small scale. Developers used Storm APIs to create jobs for stream processing. In this scenario, developers must have proficient technical skills, handle debugging challenges, and perform large amounts of repetitive work.

To address these challenges, we started working on data encapsulation and abstraction. Before data encapsulation and abstraction, we needed to choose an integrated processing engine for stream and batch processing from the available options: Apache Spark and Flink. The key difference of Apache Spark and Flink lies in the way they process data streams and batches. In Apache Spark, data streams are divided into micro batches, which are then processed by the Spark engine to generate the final stream of results in batches. For this method, the overhead must be increased to achieve a lower delay. Therefore, it is hard to reduce the delay of Spark Streaming to seconds or to sub-second level. In Apache Flink, batches are considered as bounded data streams that have a defined start and end. In this way, most code can be shared for stream and batch processing, which allows you to leverage the advantages of batch processing. Based on a thorough comparison between Apache Spark and Flink, we decided to use Apache Flink as the processing engine for real-time computations over data streams. Stream processing methods can be classified as stateful computations and stateless computations. The introduction of state management allows you to easily implement complex processing logic, which is ground-breaking for stream processing.

Any emerging technology is only adopted by a small group in the beginning. With the growth of this technology and the reduction in adoption costs, it will be widely accepted. Therefore, we are working to enable stream processing technologies to be widely adopted by improving the technology and decreasing adoption costs. Apache Flink has made many improvements to the architecture, but its implementation mechanism needs to be optimized. For example, the tasks of multiple jobs may be executed by the same thread, which greatly reduces the computing performance. To resolve this issue, we introduce the YARN system. Another example is the checkpoint feature of Apache Flink. In Apache Flink, checkpoints are created to ensure data consistency, but checkpoints cannot be created when the state stored for incremental computing is excessively large. To address this challenge, Realtime Compute optimizes the checkpoint feature to efficiently manage large state. Realtime Compute has addressed many performance issues and bottlenecks to ensure the stability and scalability in the production environment. Currently, Realtime Compute is capable of supporting core businesses. We have also improved the SQL of Realtime Compute to support complex business scenarios. We are working to provide excellent user experience through constant exploration and innovation.

23.1.2. Key challenges of Realtime Compute

Realtime Compute runs on a cluster of thousands of nodes within Alibaba Group. It provides services for hundreds of real-time applications for over 20 business units of Alibaba Group, processing hundreds of billions of messages and about 1 petabyte of traffic per day. Realtime Compute has become one of the core distributed computing services of Alibaba Group.



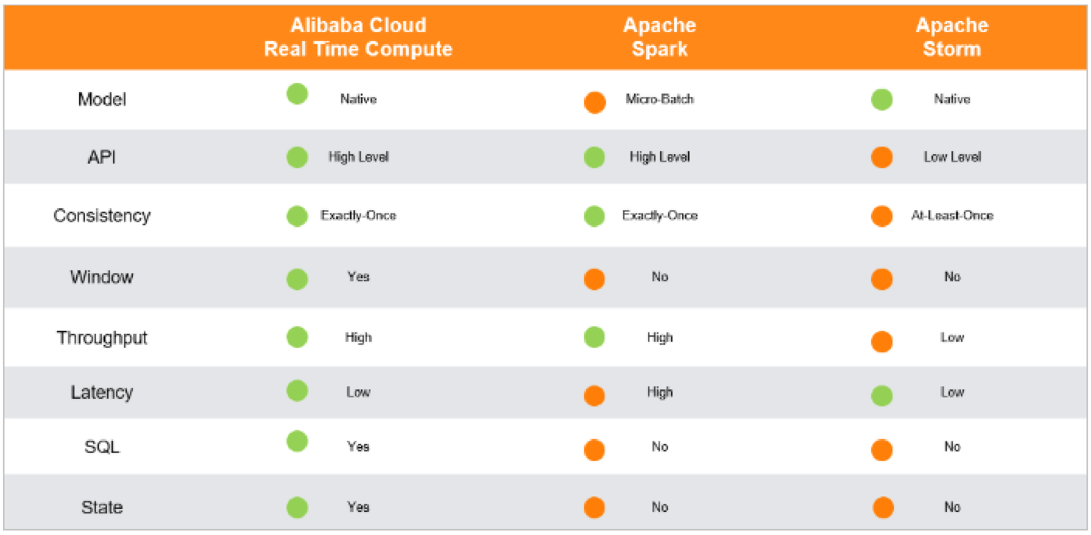
We are working to make the following improvements:

- Computing engine: We are working to improve the engine performance and enable the engine to support multiple semantics of processing messages.
- Programming interfaces: We are working to enable support for more APIs and programming languages. For example, we are working on the compatibility with open source APIs, such as Storm APIs and Beam APIs.
- Programming languages: We are working to enable support for more SQL syntaxes and semantics in stream analysis scenarios, such as temporal tables and complex event processing (CEP). Services: We are working to improve Realtime Compute from the following aspects: debugging, one-click deployment, hot upgrades, and training systems.

23.2. Benefits

Realtime Compute uses a compute engine that is developed based on Apache Flink, which allows Realtime Compute to use advantages of Apache Flink and optimize the Flink Table API. You can use Flink SQL for batch and stream processing. The application of YARN in Realtime Compute enables full compatibility with Flink API, which enables a large ecosystem of stream processing.

Realtime Compute and other stream processing systems



	Alibaba Cloud Real Time Compute	Apache Spark	Apache Storm
Model	🟢 Native	🟠 Micro-Batch	🟢 Native
API	🟢 High Level	🟢 High Level	🟠 Low Level
Consistency	🟢 Exactly-Once	🟢 Exactly-Once	🟠 At-Least-Once
Window	🟢 Yes	🟠 No	🟠 No
Throughput	🟢 High	🟢 High	🟠 Low
Latency	🟢 Low	🟠 High	🟢 Low
SQL	🟢 Yes	🟠 No	🟠 No
State	🟢 Yes	🟠 No	🟠 No

This figure shows the differences between the technologies of Realtime Compute and other stream processing systems. Based on the extensive experience of addressing challenging business scenarios, Realtime Compute provides the following benefits:

- **Powerful features**

  Unlike these open source systems, Realtime Compute simplifies the development process by integrating a wide range of features.

  - A powerful engine is used. This engine offers the following advantages:
    - Provides the standard Flink SQL that enables automatic data recovery from failures. This ensures accurate data processing when failures occur.
    - Supports multiple types of built-in functions, such as text functions, date and time functions, and statistics functions.
    - Enables an accurate control over computing resources to isolate the jobs of tenants.
  - The key performance metrics of Realtime Compute are three to four times higher than those of Apache Flink. For example, in Realtime Compute, the data processing delay is reduced to seconds or even to sub-second level. The throughput of a job reaches millions of data records per second. A cluster can contain thousands of nodes.
  - Realtime Compute integrates cloud-based data stores such as MaxCompute, DataHub, Log Service, ApsaraDB RDS, Tablestore, and AnalyticDB for MySQL. Realtime Compute allows you to read data from and write data to these systems with the least efforts in data integration.

- Managed real-time computing services

  Unlike open source or self-managed stream processing services, Realtime Compute is a fully managed stream processing engine. You can query streaming data without the need to deploy or manage the infrastructure. Realtime Compute allows you to use streaming data processing services with a few clicks. Realtime Compute integrates services such as development, administration, monitoring, and alerting. This allows you to use cost-effective streaming data services for trial and migrate your data for deployment.

  Realtime Compute also enables complete isolation between tenants. This isolation and protection extends from the top application layer to the underlying infrastructure layer. This helps to ensure the security and privacy of your data.

- Excellent user experience during development

Realtime Compute provides a standard SQL engine: Flink SQL. Realtime Compute also provides many built-in functions, such as the text functions, date and time functions, and statistics functions. The application of these functions greatly simplifies and accelerates the Flink-based development. Flink SQL allows users with limited development knowledge, such as business intelligence (BI) analysts and marketers, to easily perform real-time analysis and processing of big data.

Realtime Compute provides an end-to-end solution for stream processing, including development, administration, monitoring, and alerting. On the Realtime Compute development platform, only three steps are required to publish a job.

- Low costs

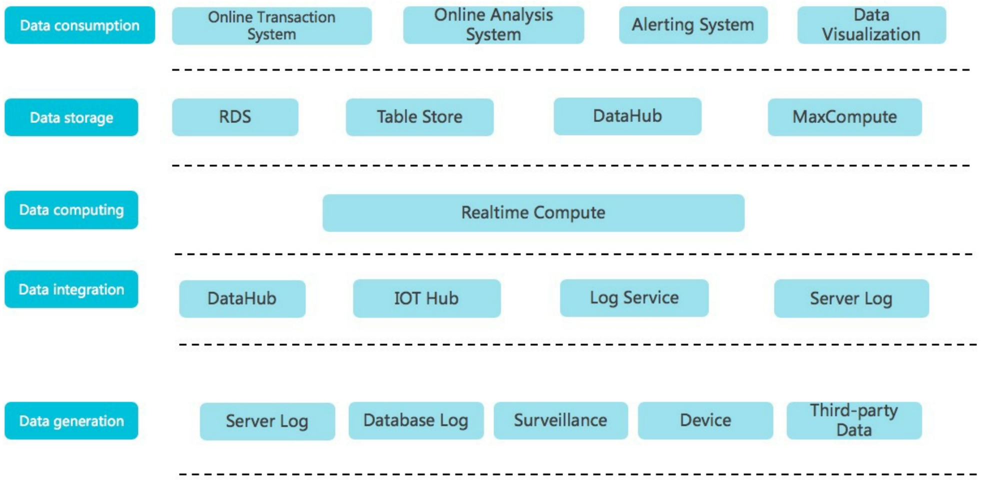
  Many improvements are made to the SQL execution engine, which allows you to create jobs more cost-effectively than to create Flink jobs. Realtime Compute is more cost-effective than open source stream frameworks in both development and production costs.

23.3. Product architecture

23.3.1. Business architecture

Realtime Compute is a lightweight SQL-enabled streaming engine for real-time processing and analysis of data streams.

Business architecture



- Data generation

  In this phase, streaming data is generated from sources such as server logs, database logs, sensors, and third-party systems. The generated streaming data moves on to the next phase for data integration to drive real-time computing.

- Data integration

  In this phase, the streaming data is integrated. You can subscribe to and publish the integrated streaming data. The following Alibaba Cloud products can be used in this phase: DataHub for big data computing, IoT Hub for connecting IoT devices, and Log Service for integrating ECS logs.

- Data computing

In this phase, the streaming data, which has been subscribed to in the data integration phase, acts as inputs to drive real-time computing in Realtime Compute.

- Data storage

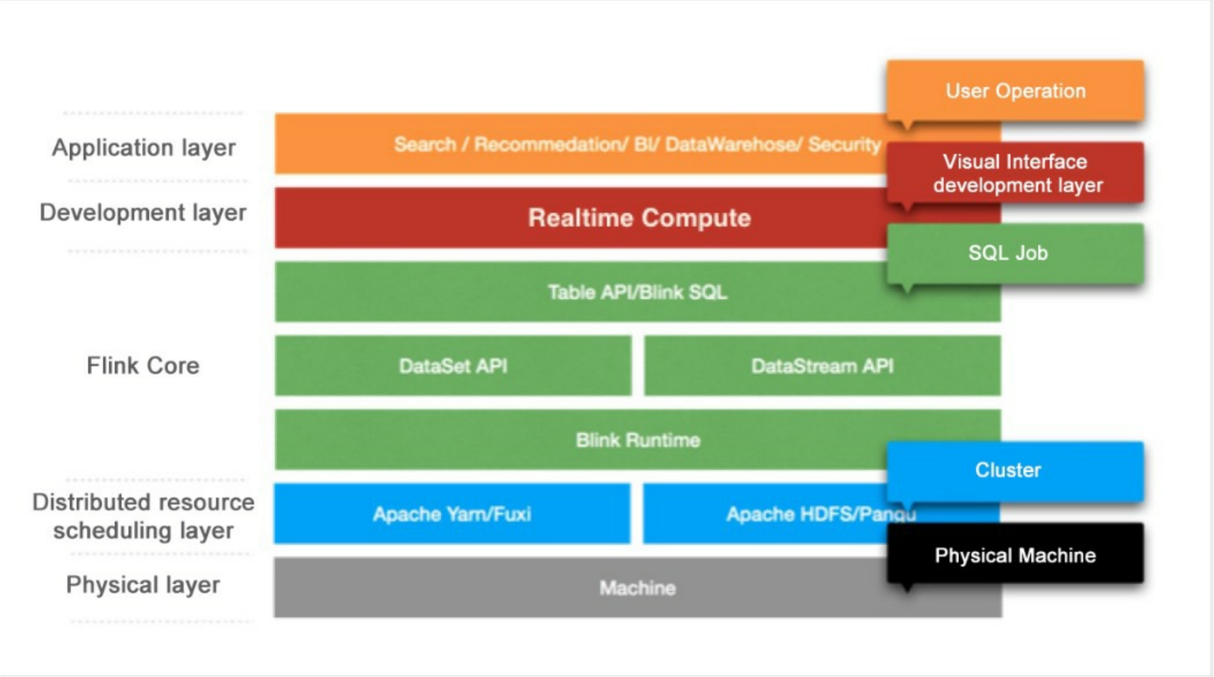
  Realtime Compute does not provide built-in data stores. Instead, it writes computing results to external data stores, such as relational databases, NoSQL databases, and online analytical processing (OLAP) systems.

- Data consumption

  Realtime Compute supports multiple data store types, which allows you to consume data in various ways. For example, data stores for message queues can be used to report alerts, and relational databases can be used to provide online support.

23.3.2. Technical architecture

Realtime Compute is a real-time data analysis platform for incremental computing. This platform provides statements that are similar to SQL statements and uses the MapReduceMerge (MRM) computing model for incremental computing. Realtime Compute offers a failover mechanism to ensure data accuracy when errors occur.



The Realtime Compute architecture consists of the following five layers.

- Application layer

  This layer allows you to create SQL files and publish jobs for real-time data processing based on a development platform. With a well-designed monitoring and alerting system, you would be notified of a processing delay for each job in a timely manner. You can also use systems like Flink UI to view the running information of published jobs and analyze performance bottlenecks. This allows you to quickly and effectively improve job performance.

- Development layer

This layer parses Flink SQL and generates logical and physical execution plans. The execution plans are then conceptualized as executable directed acyclic graphs (DAGs). Based on these DAGs, directed graphs that consist of various models are obtained. Directed graphs are used to implement specific business logic. A model usually contains the following three modules:

- Map: Operations such as data filtering, distribution (GROUP), and join (MAPJOIN) are performed.
- Reduce: Realtime Compute processes streaming data by batch, and each batch contains multiple data records.
- Merge: You can update the state by merging the computing results of the batch, which are produced from the Reduce module, with the previous state. Checkpoints are created after N (configurable) batches have been processed. In this way, the state is stored persistently in a data store, such as Tair and Apache HBase.

- Flink Core

  This layer provides a wide range of computing models, Table API, and Flink SQL. You can use DataStream API and DataSet API at the lower sublayer. At the bottom sublayer is Flink Runtime, which schedules resources to ensure that jobs can run properly.

- Distributed resource scheduling layer

  Realtime Compute clusters run based on the Gallardo scheduling system. This system ensures that Realtime Compute runs effectively and fault tolerance is provided for recovery.

- Physical layer

  This layer provides powerful hardware devices for clusters.

23.4. Functional principles

The Blink engine of Realtime Compute is developed based on Apache Flink. For more information about the functional principles of Realtime Compute, see Discussion on Apache Flink.

24.Machine Learning Platform for AI

24.1. What is Machine Learning Platform for AI?

Machine Learning Platform for AI uses statistical algorithms to train models by using a large amount of historical data and generate empirical models. You can use these models to make informed business decisions.

Machine Learning Platform for AI is a set of tools for data mining, modeling, and prediction. It is developed based on the distributed computing engine MaxCompute. Machine Learning Platform for AI provides you with the following benefits:

- Machine Learning Platform for AI provides you with an all-in-one algorithm service that supports algorithm development, sharing, model training, deployment, and monitoring.
- Machine Learning Platform for AI allows you to manage experiments on the graphic user interface (GUI) or by running commands. It is intended for data miners, analysts, algorithm developers, and data explorers.
- In Apsara Stack, Machine Learning Platform for AI runs on MaxCompute. After you deploy algorithms in MaxCompute clusters, you can call the algorithms from the Machine Learning Platform for AI console. This decouples algorithm applications from computing engines.
- Machine Learning Platform for AI provides you with bountiful algorithms and reliable technical support for you to resolve issues in various business scenarios. In the data technology (DT) era, you can use Machine Learning Platform for AI to develop data-driven business.

You can apply Machine Learning Platform for AI in the following scenarios:

- Marketing: commodity recommendation, user profiling, and targeted advertising
- Finance: credit risk prediction for loans, financial risk management, stock forecast, and gold price forecast
- Social network: analytics of key opinion leaders and relational networks
- Text processing: news classification, keyword extraction, document summarization, and text analysis
- Unstructured data processing: image classification and image text extraction based on Optical Character Recognition (OCR)
- Other scenarios: rainfall forecast and forecast of football match results

Machine Learning Platform for AI supports the following learning modes:

- Supervised learning: Each sample has an expected value. You can create a model to map input feature vectors to goal values. Typical examples of this learning mode include regression and classification.
- Unsupervised learning: Goal values are not specified for samples. This learning mode is used to discover potential principles based on the sample data, such as simple clustering.
- Reinforcement learning: This learning mode is complex. A system constantly interacts with external environments to obtain feedback and determines its own behaviors to achieve long-term optimization of objectives. Typical examples of this learning mode contain AlphaGo and autonomous vehicles.

24.2. Benefits

This topic describes the benefits of Machine Learning Platform for AI (PAI) such as distributed algorithm frameworks, efficient optimization of models and compilation, and rich and quality algorithms.

Distributed algorithm frameworks

- PAI supports three types of engines for deep learning, parameter servers, and Message Passing Interface (MPI).
- The optimized deep learning engine provides excellent performance.

Efficient optimization of models and compilation

Collaborative optimization of models and system compilation is a core technology provided by the modern heterogeneous computing infrastructure for AI computing services. PAI supports collaborative optimization of models and system compilation.

Scheduling of heterogeneous resources

Heterogeneous resources such as GPU resources are required by deep learning tasks. You can build independent clusters to schedule heterogeneous computing tasks.

Rich and quality algorithms

All PAI algorithms are developed based on practices in Alibaba Group. The algorithms have been tested by using petabytes of service data in complex business scenarios. This ensures that the algorithms are sophisticated and stable.

Open and AI-assisted development environments for model training with high elasticity (DSW)

- Environment customization based on Docker images

  A variety of container images that support commonly used machine learning frameworks, such as PyTorch and TensorFlow, are provided. You can use these images to deploy complex environments with ease. In addition, you can add software to these images and then deploy containers that meet your requirements.

- Kubernetes-based container scheduling

  Data Science Workshop (DSW) uses Kubernetes as an underlying resource platform where you can schedule and start containers.

- Work with open source development tools

  The container images contain out-of-box development tools, such as JupyterLab, VScode, and CLI, that are commonly used by machine learning developers.

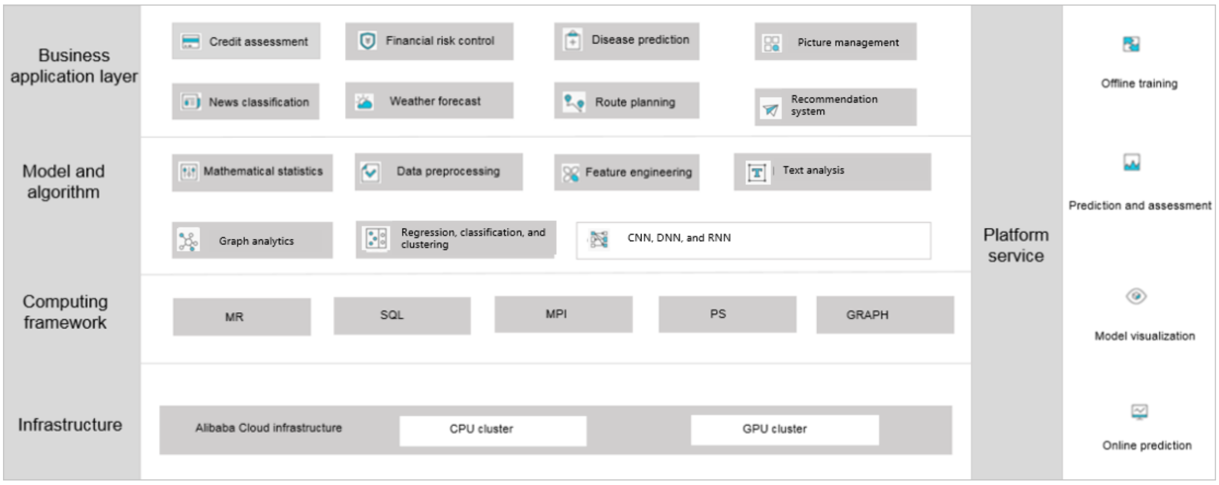
24.3. Architecture

24.3.1. System architecture

This topic describes the system architecture of Machine Learning Studio and that of Data Science Workshop (DSW) in Machine Learning Platform for AI (PAI).

System architecture of Machine Learning Studio

Machine Learning Studio provides a visualized environment where you can create models. The environment consists of multiple components. The following figure shows the system architecture of Machine Learning Studio.



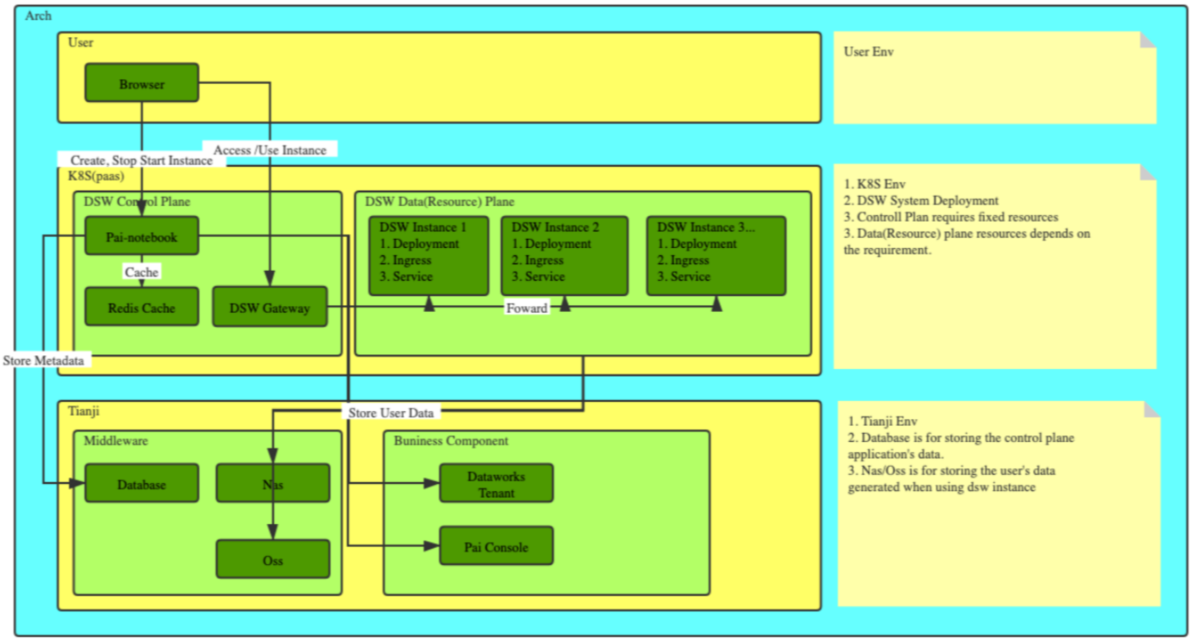
Description:

- Infrastructure layer: includes CPU and GPU clusters.
- Computing framework layer: provides computing methods such as MapReduce, SQL, and Message Passing Interface (MPI). The distributed computing architecture is used to distribute and run computing tasks in parallel.
- Model and algorithm layer: includes basic components, such as data preprocessing, feature engineering, and machine learning algorithm components. All algorithm components come from the algorithm system of Alibaba Group and have been tested by using petabytes of service data.
- Service application layer: supports different Alibaba projects, such as the search system, recommendation system, and Ant Financial, in data mining. PAI is applicable in various industries, such as finance, medical care, education, transportation, and security.

If you call models and algorithms in PAI, the system converts the algorithms into computing types. For example, if you want to join two tables, an SQL workflow is automatically generated and then delivered to MaxCompute for calculation and processing. All algorithms are stored in underlying computing engines as plug-ins. To use algorithms, you need only to call the algorithms. This decouples algorithms from computing engines.

System architecture of DSW

DSW of PAI provides an interactive environment for modeling. The system architecture of DSW consists of the control plane and data plane, as shown in the following figure.



Description:

- Control plane

  The control plane includes DSW control services, as shown in the preceding figure. You can view resource lists and perform operations on resources in the control plane. For example, you can create, modify, and delete resources and query the information about resources in the control plane. The control plane provides API operations that allow you to manipulate the data plane. For example, it provides API operations for you to start pods, deploy networks, and create Apsara File Storage NAS (NAS) resources and mount targets for Kubernetes clusters.

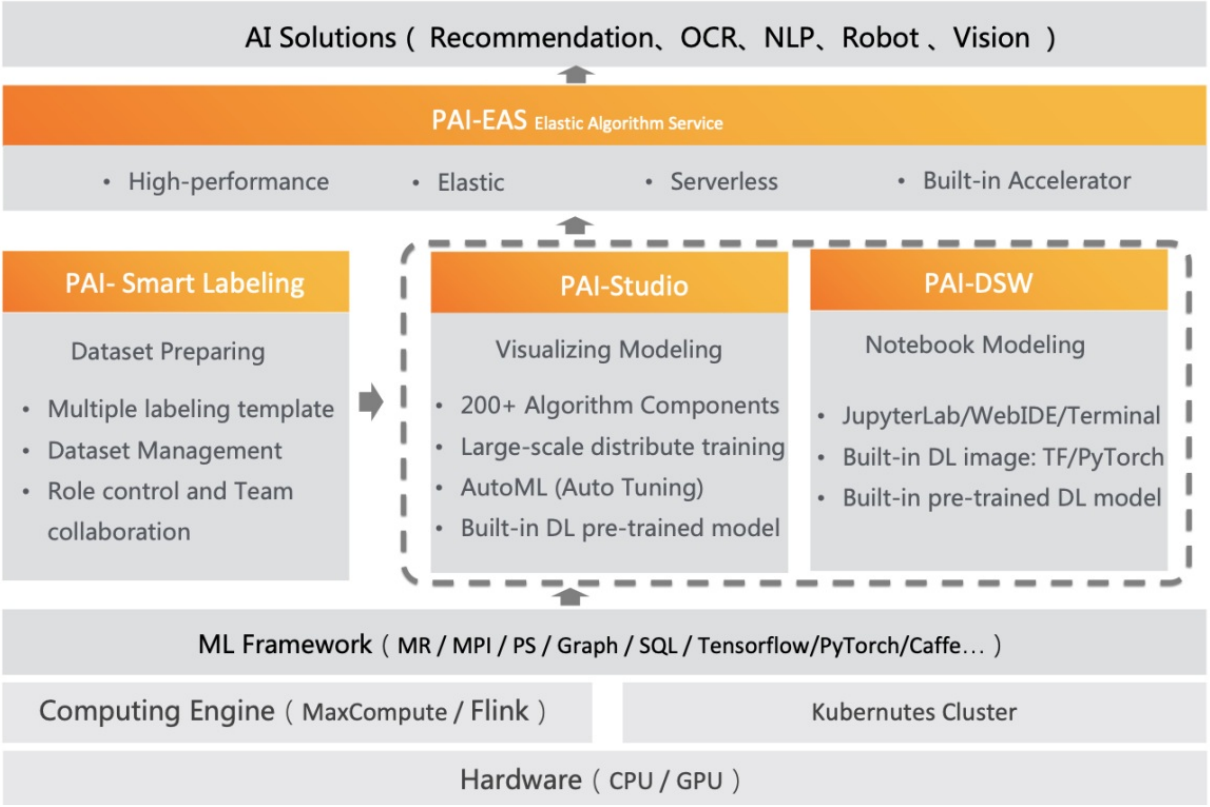
- Data plane

  The data plane includes Kubernetes clusters. Most of the DSW resources in the data plane belong to the dsw namespace. You can start pods to obtain the resources that are required to develop and run computing tasks.

24.3.2. Feature-oriented architecture

This topic describes the feature-oriented architecture of Machine Learning Platform for AI (PAI).

The following figure shows the feature-oriented architecture of PAI.


AI Solutions（ Recommendation、OCR、NLP、Robot 、Vision ）
PAI-EAS Elastic Algorithm Service
• High-performance
• Elastic
• Serverless
• Built-in Accelerator
PAI- Smart Labeling
Dataset Preparing
• Multiple labeling template
• Dataset Management
• Role control and Team collaboration
PAI-Studio
Visualizing Modeling
• 200+ Algorithm Components
• Large-scale distribute training
• AutoML (Auto Tuning)
• Built-in DL pre-trained model
PAI-DSW
Notebook Modeling
• JupyterLab/WebIDE/Terminal
• Built-in DL image: TF/PyTorch
• Built-in pre-trained DL model
ML Framework（ MR / MPI / PS / Graph / SQL / Tensorflow/PyTorch/Caffe… ）
Computing Engine（ MaxCompute / Flink ）
Kubernutes Cluster
Hardware（ CPU / GPU ）

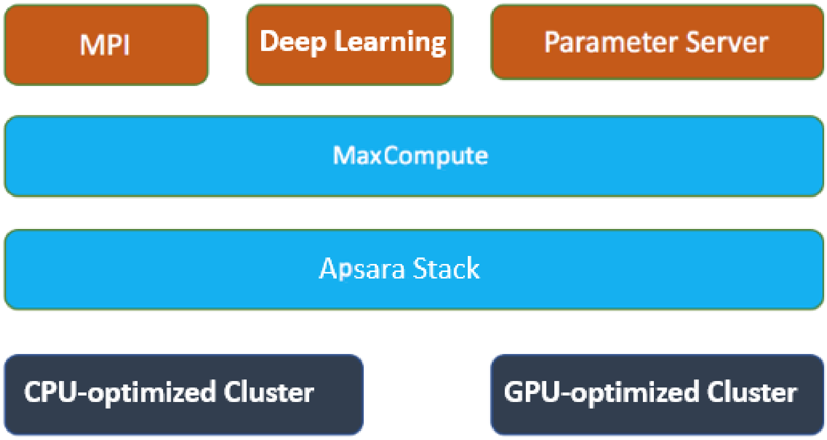
Module	Description
Infrastructure	The CPU cluster runs machine learning algorithm jobs and provides CPU and memory resources for computing. Computing resources are centrally managed by an algorithm framework. After jobs are submitted, the algorithm framework distributes the jobs to the compute nodes in the CPU cluster. The GPU cluster runs deep learning framework jobs and provides GPU and memory resources for computing. The following rules are used: ● Computing resources are centrally managed by an algorithm framework. After jobs are submitted, the algorithm framework distributes the jobs to the compute nodes in the GPU cluster. ● For a task that requires multiple multi-GPU servers, a virtual network is automatically created to distribute the jobs to the compute nodes in the virtual network.
Computing engines and platform as a service (PaaS)	PAI integrates with the computing engines in MaxCompute, Realtime Compute for Apache Flink, and Kubernetes clusters.
PAI framework	PAI uses various computing frameworks for distributed computing, such as parameter servers, TensorFlow, PyTorch, and Caffe.

Module	Description
PAI workflow	PAI streamlines the workflows of machine learning, including data preparation, model development and training, and model deployment. 1. Data preparation: provides the smart labeling feature for you to label data and manage datasets in multiple scenarios. 2. Model development and training: provides Machine Learning Studio and Data Science Workshop (DSW) for you to implement visualized modeling and create models by interactive programming. This way, different modeling requirements can be met. 3. Model deployment: provides Elastic Algorithm Service (EAS), which is the cloud-native online prediction platform of PAI, for you to deploy models as services with ease.
User business	PAI applies to various industries, such as finance, medical care, education, transportation, and security. Search systems, recommendation systems, and financial service systems of Alibaba Group all use PAI to mine data values.

24.4. Functions

24.4.1. Resource allocation and task scheduling

Artificial intelligent (AI) tasks typically consume considerable computing resources. Therefore, a distributed system is indispensable. A task must not occupy all resources or occupy a resource exclusively. Instead, a resource is shared by multiple tenants. Machine Learning Platform for AI balances the efficiency of resource usage between a single task and a cluster.

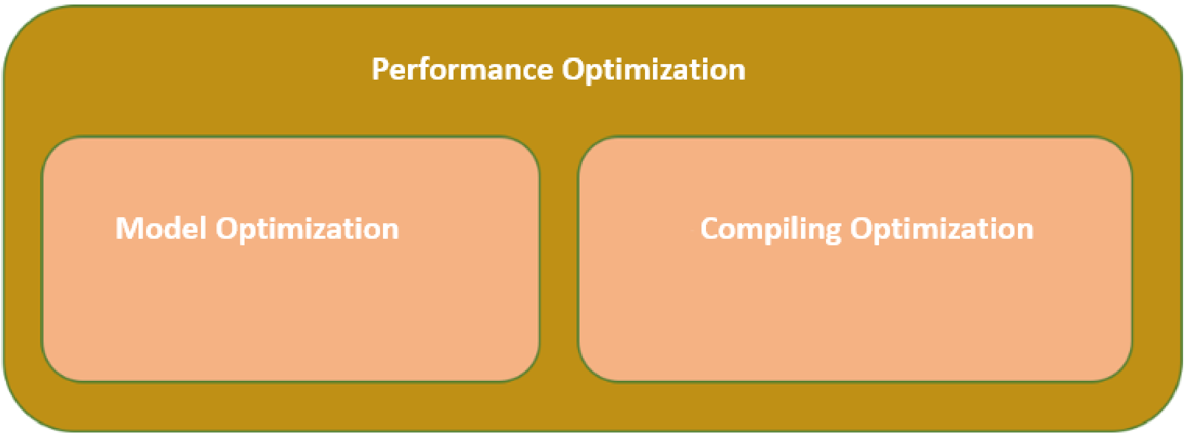


Machine Learning Platform for AI is built on the Apsara operating system and MaxCompute clusters, and is equipped with three types of compute engines: deep learning, parameter server, and MPI. AI tasks and MaxCompute tasks are deployed together to maximize the utilization of resources. For heterogeneous resources such as GPU resources required by deep learning tasks, an independent cluster is built to schedule heterogeneous computing tasks.

To allocate resources to a single task, Machine Learning Platform for AI uses the Tensorflow framework to automatically build a computing chart, allocate CPU and GPU resources, and optimize the task execution efficiency.

24.4.2. Model and compilation optimization

Collaborative optimizations of models and system compilation are a core technology provided by the modern heterogeneous computing infrastructure for AI computing services. Machine Learning Platform for AI supports the following types of optimization.



Model optimization

Many industrial service models are built based on the statistical learning theory. Model parameters can still be regularized and pruned. Besides, the AI-oriented heterogeneous computing tends to implement mixed precision to maximize the computing efficiency while guaranteeing service precision. As the hardware system develops, many technologies have been integrated in Machine Learning Platform for AI. These technologies include low bit quantization, tensor decomposition, network pruning, distillation compression, gradient compression, and hyperparameter optimization.

Compilation optimization

Model optimization aims to minimize the computing requirements when all service requirements are met. System compilation optimization is used to adapt the specified model to the heterogeneous computing architecture and release the hardware computing resources using end-to-end optimization technologies. Compilation optimization resolves the following issues:

- Computing requirement descriptions for service models. Machine Learning Platform for AI allows you to use advanced abstract languages to describe service models. You need only to describe the computing requirements. The system will translate the descriptions and perform automatic optimization.
- Hardware system independent computing chart optimization. Based on the intermediate expression of computing charts, the system implements optimizations that are independent of the hardware system structure. These optimizations include distributed splitting, mixed precision optimization, redundant computing elimination, computing mixing optimization, constant folding, efficient operator rewriting, and storage optimization of computing charts.
- Optimization and code generation related to the hardware system. The system performs optimization that is related to the hardware system and generates the target code. The optimization includes storage hierarchy optimization, parallel granularity reconstruction, computing and fetch streaming, assembly instruction optimization, and automatic CodeGen space exploration and tuning.

24.4.3. Compute engine

The compute engine provides an advanced programming language for you to compile machine learning models as needed. The engine converts the code into executable tasks at the back end, dissembles or merges the tasks, and submits the tasks to the scheduling system. Machine Learning Platform for AI supports three engines: deep learning, parameter server, and MPI.

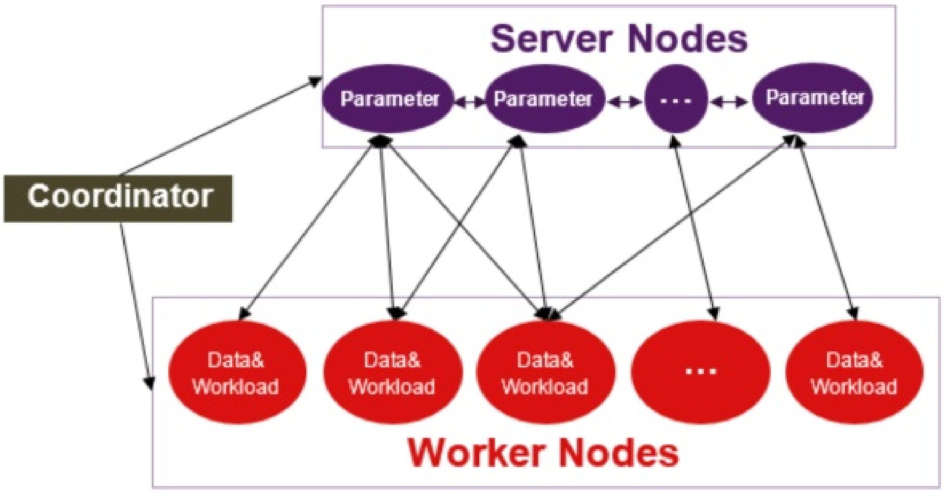
Deep learning

The deep learning engine is developed based on the open-source community TensorFlow. To adapt to the Apsara Stack cluster environment, the following improvements have been made to the deep learning engine:

- Multiple basic functions are supported. These functions include image management, service resuming, permission management, and reading and writing MaxCompute and OSS data.
- The runtime performance of the open-source TensorFlow has been improved.
  - Introduces the allreduce network primitive to improve network utilization.
  - Replaces the native gRPC mode with the RPC framework for better performance.
  - Modifies the synchronization mutex mode to reduce mutex lock competition.
- New optimizers and operators are available.

Parameter server

Parameter servers are a type of compute engine provided by Machine Learning Platform for AI for modeling training based on large models and large amounts of sample data. The engine allows algorithm developers to write distributed machine learning algorithm code in the same manner they write standalone code. Algorithm developers can implement distributed machine learning algorithms on the parameter server framework, and verify the algorithms based on tens of billions of parameter and data dimensions. This shortens the development cycle and allows new algorithms to be released for big data processing.



A parameter server supports the following functions:

- Creates hash indexes for features in real time.
- Allows you to add or delete features.
- Distributed expansion.
- Globally unified checkpoint and exactly once failover.
- Sparse hash feature-based communication.

- Embedding matrix computing based on sparse hash features.

MPI

The MPI engine is a generic distributed framework used in the industry. Machine Learning Platform for AI introduces the MPI engine and integrates the MapReduce feature of MaxCompute so that you can implement classic machine learning algorithms such as logistic regression, GBDT, FM, and K-means.

24.4.4. DSW

This topic describes how Data Science Workshop (DSW) works in the control plane and data plane. This topic also describes the network link of DSW and how computing tasks are run in DSW.

Control plane

The application logic of the control plane is managed by pai-notebook, which is a Java application that does not reside in a Kubernetes cluster. Apsara Infrastructure Management Framework is used to start the container in which pai-notebook resides. pai-notebook contains information about how to connect to external resources. Therefore, pai-notebook can be used as the middleware to manage external resources. The following external resources are included:

- Database: stores the application status of DSW, such as DSW instance information.
- Kubernetes cluster: the computing cluster that actually runs your DSW instance.
- ApsaraDB for Redis: provides cache services.
- Apsara File Storage NAS (NAS): provides persistent file storage services in remote mode.
- PAI console: provides services such as registration on web pages.
- DataWorks tenants: allows for user management operations such as logons.

When you manage the lifecycle of your DSW instance, the control plane receives the instructions that you send from the PAI console. Then, the control plane sends control instructions to backend resources to perform changes. Such a process is implemented when you create, stop, or delete a DSW instance.

Data plane

The data plane resides in the Kubernetes cluster for computing and manages the services that are provided after your DSW instance is started. The data plan contains the following types of Kubernetes resources:

- Deployment: manages your DSW instance. Each Deployment contains a replica container. The instance specifications that you select are converted to the required specifications for the container when a Deployment is enabled.
- Service: allows your DSW instance to be discovered and to provide features for external environments.
- Ingress: routes services from external environments to the internal environment of the Kubernetes cluster.

Network link

A complex network link is required for you to access the services of a DSW instance from a browser. The following network objects are included:

1. Browser
2. Oracle Parallel Server (OPS): the proxy server on Apsara Stack.

3. DSW gateway: provides authentication and HTTPS services.
4. Server Load Balancer (SLB) instance by Ingress
5. Ingress controller: routes traffic from external environments to the internal environment of the Kubernetes cluster.
6. Service: provides discovery capabilities. It is usually a virtual service. The service routes its inbound traffic to a specific pod.
7. Proxy in a pod: distributes traffic to the specified JupyterLab or Visual Studio Code service.
8. JupyterLab, Visual Studio Code, or web terminals

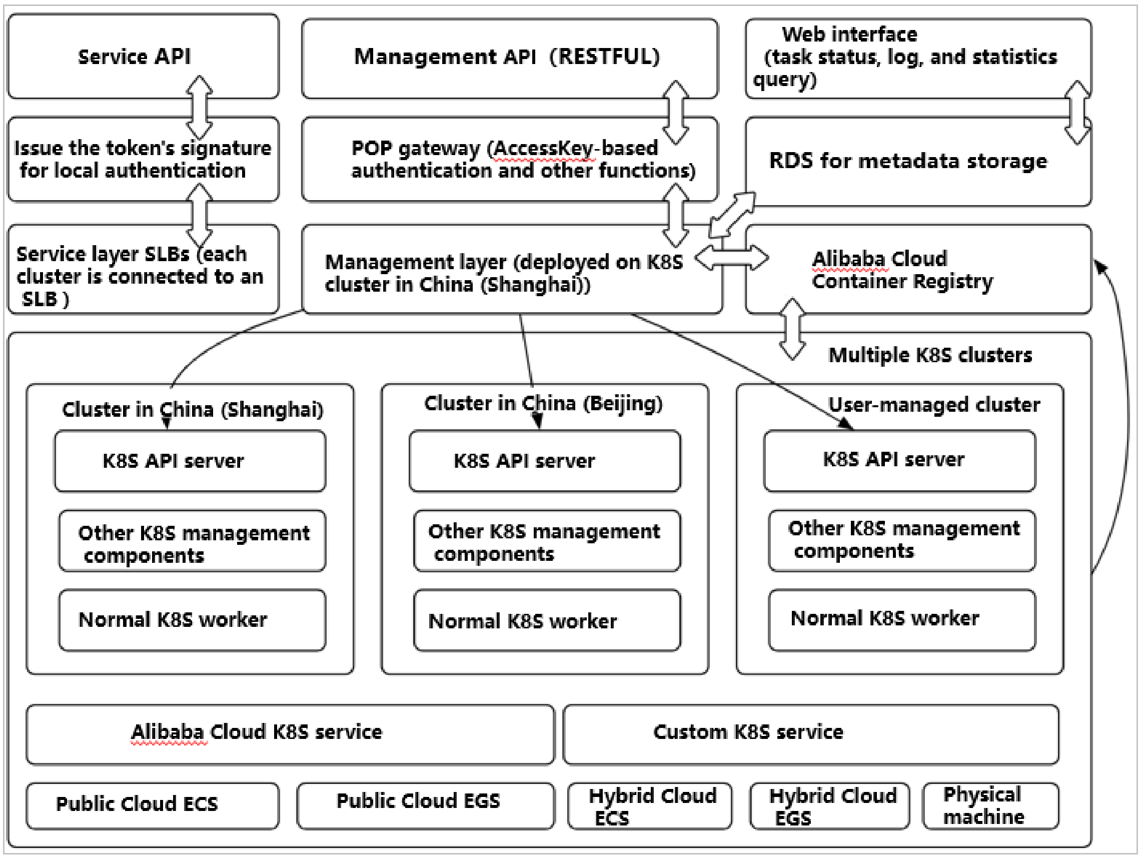
Computing tasks

If you write and run the code of a computing task by using JupyterLab or Visual Studio Code, the computing task is distributed to the container in which JupyterLab or Visual Studio Code resides. Then, the container uses the allocated computing resources to run the computing task.

24.4.5. Online prediction system

The online prediction system performs predictions tasks in the cloud using multiple types of CPUs and GPUs. The online prediction system is built based on Apsara Stack services, such as ECS, EGS, SLB, and RDS. It uses Docker to manage resources and isolate resources. It uses open-source Kubernetes (K8s) to schedule tasks.

The overall architecture of the online prediction system is as follows:


Service API
Management API (RESTFUL)
Web interface (task status, log, and statistics query)
Issue the token's signature for local authentication
POP gateway (AccessKey-based authentication and other functions)
RDS for metadata storage
Service layer SLBs (each cluster is connected to an SLB)
Management layer (deployed on K8S cluster in China (Shanghai))
Alibaba Cloud Container Registry
Multiple K8S clusters
Cluster in China (Shanghai)
K8S API server
Other K8S management components
Normal K8S worker
Cluster in China (Beijing)
K8S API server
Other K8S management components
Normal K8S worker
User-managed cluster
K8S API server
Other K8S management components
Normal K8S worker
Alibaba Cloud K8S service
Custom K8S service
Public Cloud ECS
Public Cloud EGS
Hybrid Cloud ECS
Hybrid Cloud EGS
Physical machine

The preceding figure shows the architecture of the online prediction system.

API layer

The online prediction service APIs are classified into two types:

- Prediction service APIs
- Prediction request APIs

The two API types are designed with different features to meet different requirements.

- Prediction service APIs are used to create, deploy, delete, and modify prediction services.
- Prediction request APIs are used to process prediction requests sent by clients and return prediction results.

Computing layer

- Computing resources

  All computing resources are managed by Kubernetes (K8s). Each node in a K8s cluster is an ECS instance, EGS instance, or physical server.

- Failover

  Failover depends on the failover solution provided by K8s. K8s allows you to configure a listening service for each container. For example, when the listening port is set to port 80:

  - If an IRP error occurs:

    a. Port 80 has failed the health check. K8s sets the status of the pod (container group) to Unavailable. Traffic is forwarded to another pod.

    b. When a pod is restarted, the framework initializes the pod and loads the model. Port 80 is not enabled until the model has been loaded.

    c. Port 80 is enabled after the initialization is completed. After port 80 passes the health check at the scheduling layer, the pod is added to the traffic pool again.

  - If a node in a K8s cluster fails: The keepalive message exchange between the K8s primary node and the failed node fails. K8s sets the status of the failed node to Not Ready. The pod (container group) running on the node is migrated to another node. The traffic is also forwarded to another node.

- Rolling update

  Rolling updates indicate application updates with zero downtime. The updates are classified into two types:

  - User data update: User data about the model and processor is updated using an API provided by the online prediction service. The back end creates a new image version based on the current image version and updates the deployment.
  - IRP framework code update: The framework code is updated by creating a procedure to update all user tasks in the back end. Framework code update follows the rolling update procedure. Users are not aware of the update procedure.

  User data and framework code are decoupled during cluster scheduling and they can be updated separately. The system packages user data and framework code into images of later versions separately and then modifies the description file of the existing application deployment. K8s performs rolling updates for running pods and dynamically switches the traffic to ensure that users are unaware of ongoing updates.

24.4.6. List of functions by module

Machine Learning Platform for AI provides a complete workflow of machine learning, such as data uploading, data processing, data visualization, model training, model deployment, model evaluation, and model utilization.

The following table describes the modules and corresponding functions.

Module	Function	Description
Data control	Data uploading	You can upload data through Machine Learning Platform for AI. When you upload data, the data is parsed, verified, and any errors are recorded and reported.
	Data table displaying	On Apsara Stack Machine Learning Platform for AI, click **Data Source** in the left-side navigation pane to view the uploaded data tables. You can enter a data table name in the search box and click the search icon to search for a data table. Fuzzy search is also supported.
	Data visualization	Right-click a component and choose **View Data** from the shortcut menu to view data in histograms, pie charts, or line charts.
Model control	Model training	On Machine Learning Platform for AI, click **Run** in the upper section of the canvas to train and generate a model.
	Model visualization	On Machine Learning Platform for AI, click **Models** in the left-side navigation pane. Right-click a model and choose **Show Model** from the shortcut menu to view model parameters. Tree models and linear models can be displayed in tables.
	Model downloading	Right-click a model and choose **Export PMML** from the shortcut menu to generate and download a PMML file. A PMML is a standard model description file which can be parsed by a variety of open-source software.
	Model-based prediction	You can connect model generation components and prediction components. The system will automatically use the generated model for prediction.
	Model addition, deletion, modification, and query	Right-click a model and choose to add, delete, modify, or query a model.
	Online model service	You can use the online model service to deploy a model and call the corresponding RESTful API for online prediction.

Module	Function	Description
	DataWorks task scheduling	You can deploy experiments to DataStudio as DataWorks tasks and configure the system to periodically run the tasks.
	Model evaluation	You can evaluate models using confusion matrix, binary classification evaluation, clustering model evaluation, and regression model evaluation. Models are evaluated based on metrics such as F1 score, AUC, and KS. All evaluation results can be viewed in tables or charts.
Experiment control	Whole experiment lifecycle control	You can add, delete, modify, query, and copy experiments.
	Experiment visualization	Animated visualizations are used to display the entire procedure by which an experiment runs.
	Notifications	The status of a running experiment is displayed in a prompt in the upper-right corner of the canvas, such as success and error messages.
Deep learning	Multiple deep learning frameworks	Three mainstream deep learning frameworks are supported: TensorFlow, Caffe, and MXNet. With many underlying optimizations, TensorFlow delivers better performance than other open-source frameworks.
	TensorBoard	You can view the training status of each layer in a TensorBoard job in real time and display the results visually.
	Automatic authorization	When the data source of a TensorFlow project is set to OSS, you must obtain permissions on OSS before you can run an experiment. Machine Learning Platform for AI supports automatic authorization, allowing you to obtain the read and write permissions on OSS with a single click.
	Visualized TensorFlow execution settings	The TensorFlow component is added to provide related data source settings, allowing you to run the component visually. On the Tuning tab, you can specify the number of GPUs to run with and implement parallel training with multiple GPUs easily.
	Scheduling	Deep learning jobs can be deployed and periodically executed in DataWorks.
Dashboard	Experiment history chart	You can view the experiment history on the dashboard page.
	Running experiments	You can view running experiments or delete a running experiment to save resources.

Module	Function	Description
	Scheduled tasks	You can view scheduled tasks that have been deployed and add, delete, modify, and query tasks through DataStudio.
Templates on the homepage	Machine Learning Platform for AI provides many built-in experiment templates	The experiment templates can be used for a wide range of scenarios such as product recommendations, news classification, financial risk control, haze prediction, heart disease prediction, agricultural loan delivery, and census. All these cases contain complete data sets and instructions about their use. You can also create your own experiments by using these templates.
Online prediction	Model version management	You can upload multiple versions of a model, configure them to share the same resources, and switch between those versions.
	Blue-green model deployment	The blue-green model deployment function allows you to dynamically change the proportions of the traffic forwarded between different versions of a model.
	Online model debugging	The online debugging function of Machine Learning Platform for AI allows you to debug deployed models online and view the debugging results in real time.

24.5. System metrics

Metric	Requirement
Core metrics	Provides typical machine learning algorithms, such as the data preprocessing, feature engineering, statistical analysis, classification, regression, and clustering: • Provides model evaluation algorithms. • Provides time series, text analysis, and network analysis algorithms. • Provides deep learning frameworks such as TensorFlow. • Provides the GPU job scheduling capability. • Provides the online model service and allows you to directly deploy models to the online model service. • Provides a visual console to help you use visual components to create experiments.
	Supports reading structured and unstructured data.

Metric	Requirement
Function metrics	• Supports data sampling and filtering algorithms, such as the random sampling, weighted sampling, and stratified sampling. • Supports data merging algorithms, such as JOIN, UNION, and MERGE. • Supports data preprocessing algorithms, such as splitting, normalization, standardization, KV to Table, Table to KV, and adding ID columns to tables.
	• Supports the principal component analysis (PCA) algorithm. • Supports feature importance evaluation for linear and random forest models.
	Supports the following statistical analysis algorithms: the covariance, empirical probability density chart, whole table statistics, chi-square goodness of fit test, chi-square test of independence, scatter plot, correlation coefficient matrix, two sample T test, single sample T test, normality test, percentile, Pearson coefficient, and histogram.
	• Supports the following binary classification algorithms: the Gradient Boosting Decision Tree (GBDT), Linear Support Vector Machine (SVM), and logistic regression. • Supports the following multiclass classification algorithms: K-nearest neighbors (KNN), multiclass classification for logistic regression, random forest, and naive Bayes. • Supports the GBDT, linear regression, PS-SMART regression, and PS linear regression algorithms. • Supports K-means clustering. • Supports the following evaluation algorithms: the binary classification model, regression model, clustering model, multiclass classification, and confusion matrix.
	• Supports the deep learning framework TensorFlow. • Supports TensorBoard. • Supports scheduling a deep-learning job to a GPU server.
	Supports time series algorithms such as x13_arima and x13_auto_arima.
	Supports the following text algorithms: the word frequency statistics, TF-IDF, parallel latent dirichlet allocation (PLDA), Word2Vec, word splitting, converting rows, columns, and values to KV pairs, string similarity, deprecated word filtering, text summarization, document similarity, sentence splitting, keyword extraction, ngram-count, semantic vector distance, and pointwise mutual information (PMI).
	Supports the following network analysis algorithms: the K-Core, single-source shortest path, page rank, label propagation clustering, label propagation classification, modularity, maximum connected subgraph, vertex clustering coefficient, edge clustering coefficient, counting triangle, and tree depth.

Metric	Requirement
	• Supports the online model service and allows you to deploy machine learning algorithm models or deep-learning models to the service. • Provides an HTTP-based API.
	• Provides the Web-based visual editor, which allows you to create an experiment by dragging and dropping components. • Supports releasing experiments to DataWorks for task scheduling. • Supports experiment and model management.
Compatibility/openness	• Supports the open-source deep learning framework TensorFlow 1.4. • Supports exporting the PMML file from machine learning models. • Supports the online service model and allows you to deploy the model as an API.

25.DataHub

25.1. What is DataHub?

25.1.1. Overview

DataHub collects, stores, and processes streaming data, allowing you to analyze streaming data and build applications based on the streaming data.

DataHub is a platform 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.

DataHub collects, stores, and processes streaming data from mobile devices, applications, website services, and sensors. You can use your own applications or Apsara Stack Realtime Compute to process streaming data in DataHub, such as real-time website access logs, application logs, and events. The processing results such as alerts and statistics presented in graphs and tables are updated in real time.

Based on the Apsara system of Alibaba Cloud, DataHub features high availability, low latency, high scalability, and high throughput. DataHub is seamlessly integrated with Realtime Compute, allowing you to use SQL to analyze streaming data.

DataHub can also distribute streaming data to Apsara Stack services such as MaxCompute and Object Storage Service (OSS).

DataHub supports the following features:

- **Data queue**: DataHub automatically generates a cursor for each record in a shard, which can be considered as a logical data queue. The cursor is a unique sequence of numbers. You can improve the performance of a topic by increasing the number of shards in the topic.
- **Offset-based data consumption**: DataHub saves consumption offsets for applications. You can resume data consumption from a saved consumption offset when your application fails.
- **Data synchronization**: Data in DataHub can be automatically synchronized to other Apsara Stack services, including MaxCompute, OSS, AnalyticDB, ApsaraDB RDS for MySQL, Tablestore, and Elasticsearch.
- **Scalable topics**: DataHub allows you to scale in or out topics by splitting or merging shards.

25.1.2. Benefits

High throughput

You can write terabytes (TB) of data into a topic and up to 80 million records into a shard every day.

Real-time processing

DataHub makes it easy to collect and process various types of streaming data in real time so you can react quickly to new information.

Ease of use

- DataHub provides a variety of SDKs for C++, Java, Python, Ruby, and Go.
- In addition to SDKs, DataHub provides RESTful APIs so that you can manage DataHub by using existing protocols.

- You can use collection tools such as Fluentd, Logstash, and Oracle GoldenGate to write streaming data into DataHub.
- DataHub supports structured and unstructured data. You can write unstructured data to DataHub, or create a schema for the data before it is written into the system.

High availability

- The processing capacity of DataHub is automatically scaled out without affecting your services.
- DataHub automatically stores multiple copies of data.

Scalability

You can dynamically adjust the throughput of each topic. The maximum throughput of a topic is 256,000 records per second.

Data security

- DataHub provides enterprise-level security measures and isolates resources between users.
- It also provides several authentication and authorization methods, including whitelist configuration and RAM user management.

25.1.3. Highlights

The highlights of DataHub features are described as follows:

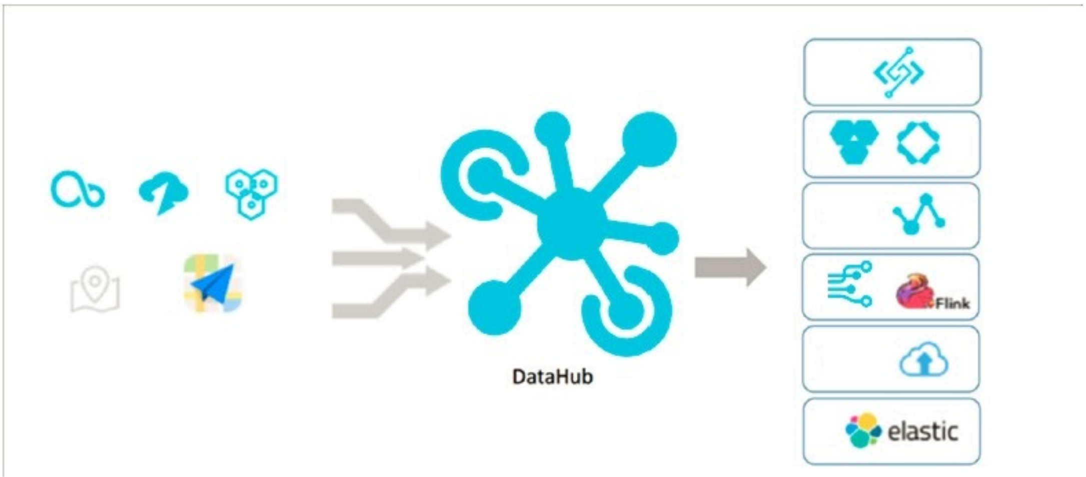
Highlights

Highlight	Description
Data security	DataHub ensures data security based on the Alibaba Cloud RAM system.
Simple O&M	DataHub automatically deactivates and recovers problematic nodes before reactivating the nodes.
Resource isolation	DataHub isolates resources between tenants.
Connection with various Alibaba Cloud services	DataHub can be used with a variety of other Alibaba Cloud services.
Scalability	The processing capacity of DataHub is automatically expanded without affecting your services. The scalability is verified during the service peak of Double 11.
Read/write performance	Records written into DataHub can be consumed repeatedly within the time-to-live of the records.
High availability	DataHub offers various high availability solutions.
Seamless integration with Alibaba Cloud services	DataHub is seamlessly integrated with various Alibaba Cloud services.

25.1.4. Scenarios

Data uploading

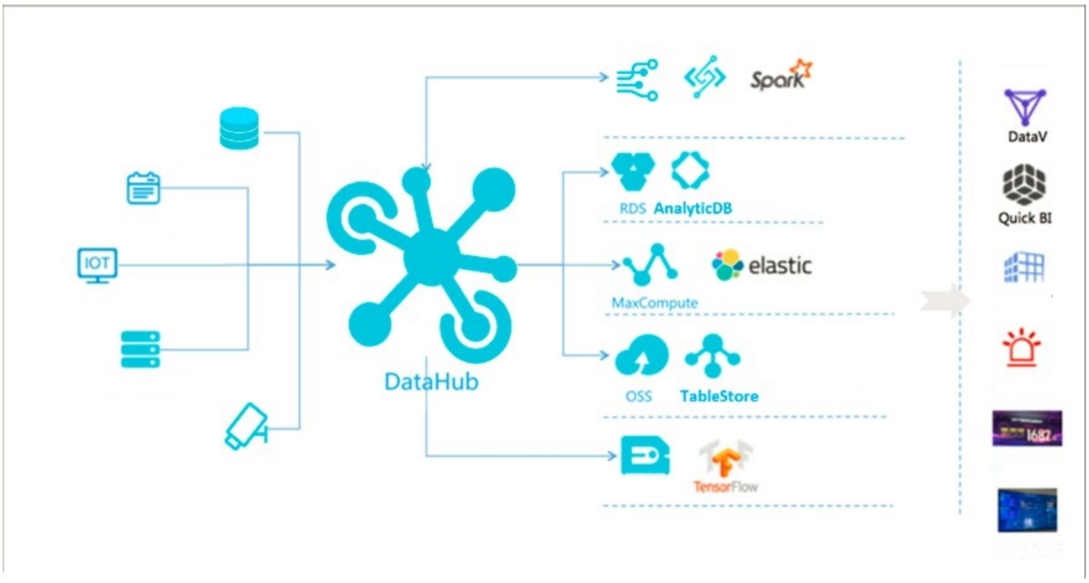
Data uploading



DataHub is connected to other Alibaba Cloud services, saving you the trouble of uploading the same data to different platforms.

Data collection

Data collection



DataHub provides several types of data collection tools for you to write your data into DataHub. DataHub supports log collection from Logstash and Fluentd, and binary log collection from Data Transmission Service (DTS) and Oracle GoldenGate (OGG). DataHub also supports the collection of surveillance videos through GB28181.

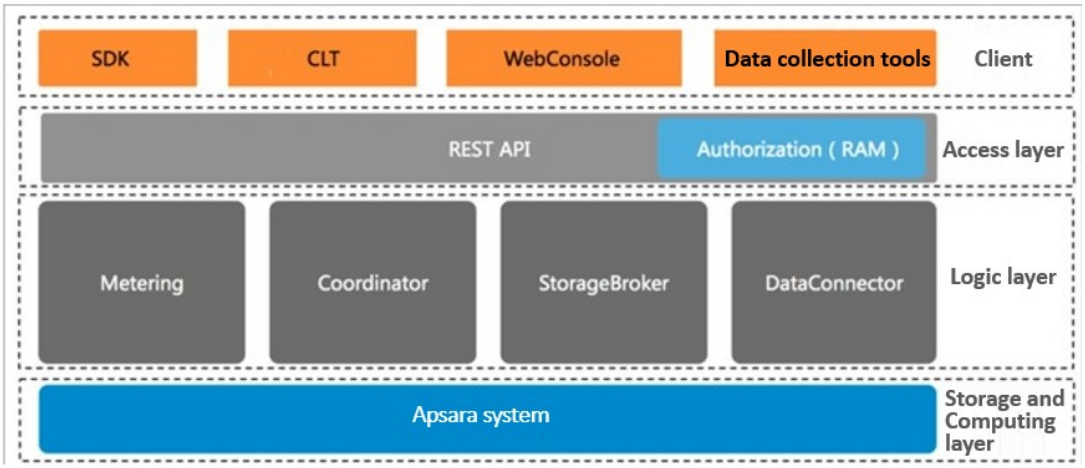
25.2. Architecture

25.2.1. Feature oriented architecture

Feature oriented architecture of DataHub shows the feature oriented architecture of DataHub.

Feature oriented architecture of DataHub

Feature oriented architecture of DataHub



The architecture of DataHub consists of four layers: **clients**, **access layer**, **logic layer**, and **storage and scheduling layer**.

Clients

DataHub supports the following types of clients:

- SDKs: DataHub provides SDKs in a variety of languages such as C++, Java, Python, Ruby, and Go.
- Command-line tools (CLTs): You can run commands in Windows, Linux, or Mac operating systems to manage projects and topics.
- Console: In the console, you can manage projects and topics, create subscriptions, view the shard status, monitor topic performance, and manage DataConnectors.
- Data collection tools: You can use Logstash, Fluentd, and Oracle GoldenGate (OGG) to collect data to DataHub.

Access layer

You can access DataHub by using HTTP and HTTPS. DataHub supports Resource Access Management (RAM) authorization and horizontal scaling of topic performance.

Logic layer

The logic layer handles the key features of DataHub, including project and topic management, data read and write, offset-based data consumption, traffic statistics, and data synchronization. Based on these key features, the logic layer is composed of the following modules: StorageBroker, Metering, Coordinator, and DataConnector.

- StorageBroker: provides data reads and writes in DataHub. This module adopts the log file storage model of Apsara Distributed File System, halving the read/write volume compared with the conventional write-ahead logging (WAL) model. This module stores three copies of data to ensure that no data is lost if a server fault occurs, and supports disaster recovery between data centers. It supports real-time data caching to ensure efficient consumption of real-time data and supports an independent read cache of historical data to enable concurrent consumption of historical data.
- Metering: supports shard-level billing based on the consumption period.
- Coordinator: supports offset-based data consumption and horizontal scaling of the processing capacity. It supports up to 150,000 QPS on a single node.
- DataConnector: supports automatic data synchronization from DataHub to other Apsara Stack

services, including MaxCompute, OSS, AnalyticDB, ApsaraDB RDS for MySQL, Tablestore, and Elasticsearch.

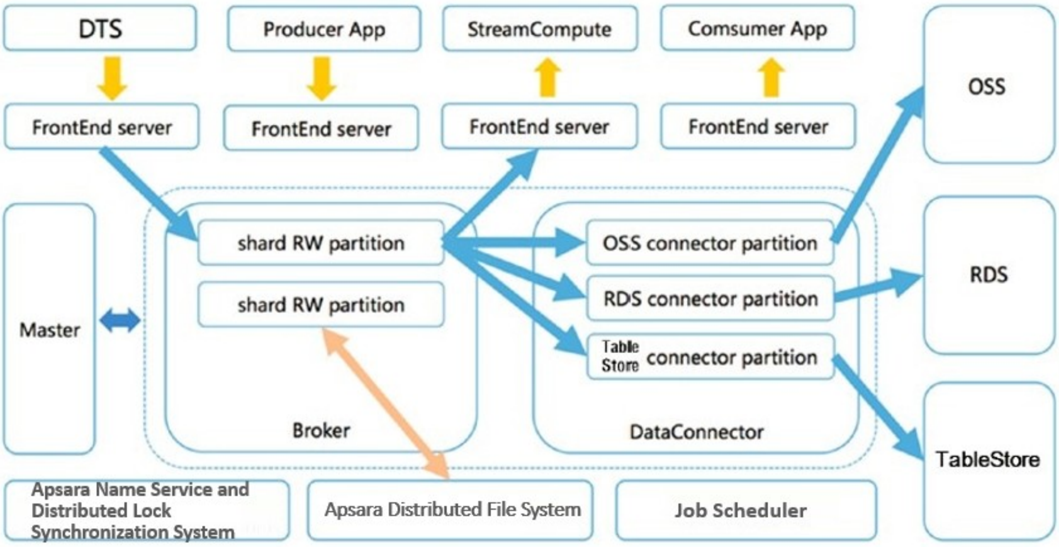
Storage and scheduling layer

- Storage: Based on the log file storage model of Apsara Distributed File System, DataHub supports append operations and solid state drive (SSD) storage. Data in each shard is stored in a separate file based on the timestamp of the data.
- Scheduling: Based on Job Scheduler, DataHub assigns shards to nodes based on the traffic on each shard. This ensures that the shards do not occupy the CPU or memory of Job Scheduler. The number of partitions on a single node has no upper limit. DataHub supports failovers within milliseconds and hot upgrades.

25.2.2. Technical architecture

Technical architecture of DataHub shows the technical architecture of DataHub.

Technical architecture of DataHub



The figure shows the process from data ingestion to consumption.

1. A shard is the smallest unit of data management in DataHub, and is a first-in, first-out (FIFO) collection of records.
2. Data in each shard is stored in a set of log files in Apsara Distributed File System.
3. The master distributes each shard to a StorageBroker. Each StorageBroker is responsible for the read and write operations on multiple shards.
4. The frontend server finds a StorageBroker based on the project, topic, and shard information specified in the request and forwards the request to the StorageBroker.
5. DataConnectors read data from the StorageBroker and forward the data to other Apsara Stack services.

Data collection

You can write data to DataHub from applications developed by using SDKs and from data collection tools such as Logstash, Fluentd, and OGG. You can also write data by using Data Transmission Service (DTS) and Realtime Compute.

Frontend server

Frontend servers constitute the access layer and support horizontal scaling. You can call RESTful API operations to access DataHub. RAM authorization is supported.

Master

The master handles metadata management and shard scheduling. It supports create, read, update, and delete operations on projects and topics. The master also supports split and merge operations on shards.

StorageBroker

StorageBrokers handle read and write operations on each shard including data indexing, caching, and file organization and management.

DataConnector

DataConnectors forward data in DataHub to other Apsara Stack services. DataConnectors provide different features for various destination services. These features include automatically creating partitions in MaxCompute and converting data streams into files stored in OSS.

25.3. Features

25.3.1. Data queue

DataHub automatically generates a cursor for each record in a shard. The cursor is a unique sequence of numbers. You can improve the performance of a topic by increasing the number shards in the topic.

25.3.2. Checkpoint-based data restoration

DataHub supports saving checkpoints for subscribed applications in the system. You can restore data from any checkpoint you saved if your subscribed application fails.

25.3.3. Data synchronization

Data in DataHub is automatically synchronized to other Alibaba Cloud services.

DataConnector

You can create a DataConnector to synchronize DataHub data in real time or near real time to other Alibaba Cloud services, such as MaxCompute, Object Storage Service (OSS), Elasticsearch, ApsaraDB RDS for MySQL, AnalyticDB, and Tablestore.

You can configure the DataConnector so that the data you write to DataHub can be used in other Alibaba Cloud services. At-least-once semantics is applied in data synchronization. This ensures that no data is lost, but may result in duplicated records in the destination platform if an error occurs during the synchronization process.

Destination platforms

The following table describes the platforms to which DataHub records can be synchronized.

Destination platforms

Destination platform	Timeliness	Description
MaxCompute	Near real-time. Latency: 5 minutes.	The column names and data types in the source topic must be the same as those in MaxCompute. The MaxCompute table must have one or more corresponding partition columns.
OSS	Real-time.	Records are synchronized to the specified bucket in OSS and are saved as CSV files.
Elasticsearch	Real-time.	Records are synchronized to the specified index in Elasticsearch. Records may not be synchronized in the order of the recording time. If you want to synchronize data in the order of the recording time, you must write the records with the same partition key into the same shard.
ApsaraDB RDS for MySQL	Real-time.	Records are synchronized to the specified table in ApsaraDB RDS for MySQL.
AnalyticDB	Real-time.	Records are synchronized to the specified table in AnalyticDB.
Tablestore	Real-time.	Records are synchronized to the specified table in Tablestore.

25.3.4. Scalability

The throughput of each topic can be scaled by splitting or merging shards.

You can adjust the number of shards in a topic according to the service load.

For example, if the topic throughput cannot handle a surge in the service load during Double 11, you can split existing shards to up to 256 to increase the throughput to 256 MB/s.

As the service load decreases after Double 11, you can reduce the number of shards as needed by performing the merge operation.

26.Apsara Big Data Manager (ABM)

26.1. What is Apsara Big Data Manager?

Apsara Big Data Manager (ABM) is an operations and maintenance (O&M) platform tailored for big data services.

ABM supports the following services:

- MaxCompute
- DataWorks
- RealtimeCompute
- Quick BI
- DataHub
- Machine Learning Platform for AI

ABM supports O&M on big data services from the perspectives of business, services, clusters, and hosts. ABM also allows you to update big data services, customize alert configurations, and view the O&M history.

Onsite Apsara Stack engineers can use ABM to easily manage big data services. For example, they can view metrics, check and handle alerts, and modify configurations.

26.2. Benefits

Apsara Big Data Manager (ABM) allows you to quickly connect to big data services and provides comprehensive O&M capabilities for each service. ABM is based on a mature O&M mid-end.

O&M mid-end

In the O&M mid-end, ABM provides various built-in services and SDKs to deliver all-around O&M capabilities. Each product can easily connect to ABM and has an exclusive site to implement O&M. Compared with the traditional development process, ABM implements a visualized, configuration-based, and function-based development process and minimizes the development costs of business customization.

The O&M mid-end provides the following services in Apsara Stack:

- Job platform: allows you to manage, run, and schedule jobs in a visualized manner. This satisfies various needs of visualized O&M.
- Knowledge graph: allows you to store, integrate, and query data in different scenarios. This service helps you easily integrate and query dispersed data.
- Function as a service (FaaS): allows you to trial and find errors at low costs, quickly develop code, and manage business logic based on functions. This service relieves you from complex project organization, dependency management, deployment, and scaling so that you can focus on your business development.
- Application management: allows you to store business logic and configurations in a hierarchical way. This service supports highly flexible extensions and allows you to create complex application structures with simple configurations by using JSON.
- Inspection: allows you to manage metrics and schedule monitoring tasks by using a universal solution. This service supports disparate alert data sources and can be embedded into any page of an

application site.

- Third-party system adaptation: allows you to use one SDK to call APIs of all the connected third-party systems.
- Authorization proxy: adapts to Average Active Sessions (AAS) and Operation Administration and Maintenance (OAM) in Apsara Stack, provides capabilities such as visualized user management and permission management, and satisfies the authorization and authentication requirements of third-party systems.
- Gateway: integrates all service APIs so that external systems can call these APIs to access the services. This service also provides isolation, decoupling, and scaffold capabilities to authenticate and process all requests in a centralized manner.
- Apsara Infrastructure Management Framework synchronization: adapts to the Apsara Infrastructure Management Framework base in Apsara Stack, and provides encapsulated interfaces for querying and managing all host data.
- Tunnel: uses StarAgent to shield the differences of underlying command execution tunnels and provides a universal interface. This allows users to deliver commands and files to a large number of hosts, and aggregate and query the statuses of these hosts.

Quick service development

ABM is based on the O&M mid-end. ABM supports multiple services such as MaxCompute and DataWorks, and provides stable and reliable O&M capabilities for these services.

26.3. Architecture

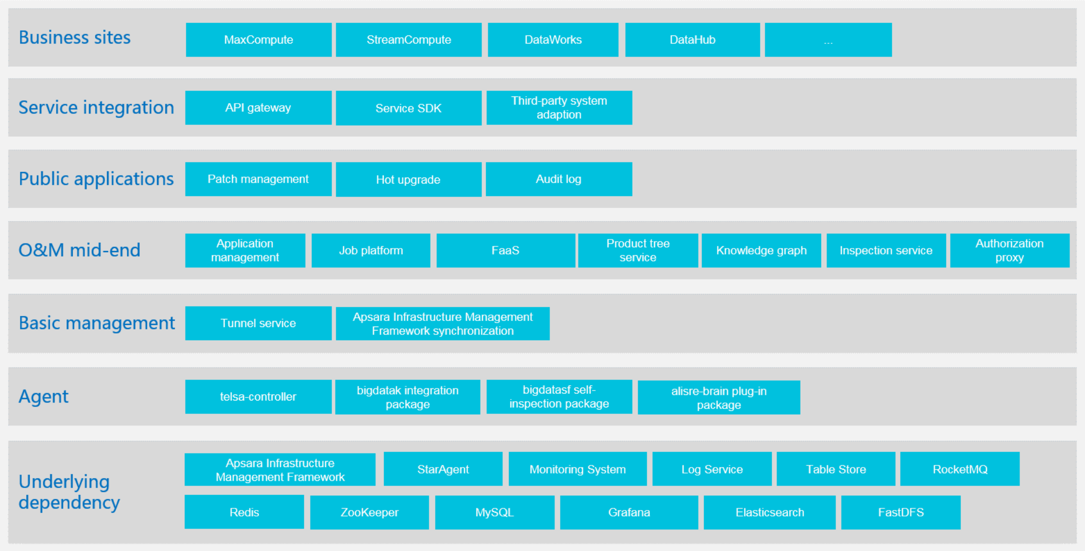
26.3.1. O&M Architecture

This topic describes the O&M architecture of Apsara Big Data Manager (ABM) and the features of each component.

ABM uses a microservice architecture that supports data integration, interface integration, and feature integration, and provides standard service interfaces. This architecture enables consistent user interfaces and O&M operations for all services in the ABM console. This reduces training costs and lowers O&M risks.

The ABM system consists of the following components: underlying dependency, agent, basic management, O&M mid-end, public applications, service integration, and business sites.

Architecture



Underlying dependency

ABM depends on Alibaba services and open source systems from third parties.

- ABM uses StarAgent and Monitoring System of Alibaba to run remote commands and execute remote data collection instructions.
- ABM uses ZooKeeper to coordinate primary and secondary services to ensure the availability of services.
- ABM uses ApsaraDB RDS to store metadata, ApsaraDB for Redis to store cache data, and Tablestore to store large amounts of self-test data. This improves service throughput.

Agent

Agent provides client SDKs, scripts, and monitoring packages that are deployed on managed servers.

O&M mid-end and basic management

The O&M mid-end and basic management components are key to ABM. Each service provides its general capabilities for business sites. This enables quick construction of business sites and makes the capabilities of each business site complete.

Public applications

Public applications are developed based on the O&M mid-end and designed with special purposes. These applications are adaptive to all big data services supported by ABM.

Service integration

Service integration links business sites with underlying components. It integrates interfaces of all internal services, adapts to various third-party systems, and provides unified SDKs for users.

Business sites

Business sites are built based on the O&M mid-end and cover all big data services such as MaxCompute, Realtime Compute for Apache Flink, DataWorks, and DataHub. Each business site provides end-to-end O&M capabilities for a service.

26.4. Features

26.4.1. Small file merging

This topic describes the small file merging feature of ABM for MaxCompute.

What are small files

Apsara Distributed File System stores data in blocks. The size of each block is 64 MB. Small files in this topic refer to files whose size is less than 64 MB. Reduce computing or real-time data collection through tunnels will generate a large number of small files.

Impacts of small files

- More small files consume more instance resources. In MaxCompute, a single task instance can handle up to 120 small files. Therefore, too many small files cause a resource waste and deteriorate system performance.
- Too many small files cause high pressure on Apsara Distributed File System, and decrease the utilization rate of disk space.
- Too many small files occupy a large amount of memories of Master servers and Chunkservers in Apsara Distributed File System. When the memory usage exceeds 50% of the safety limit on a Master server of Apsara Distributed File System, the cluster stability is affected.

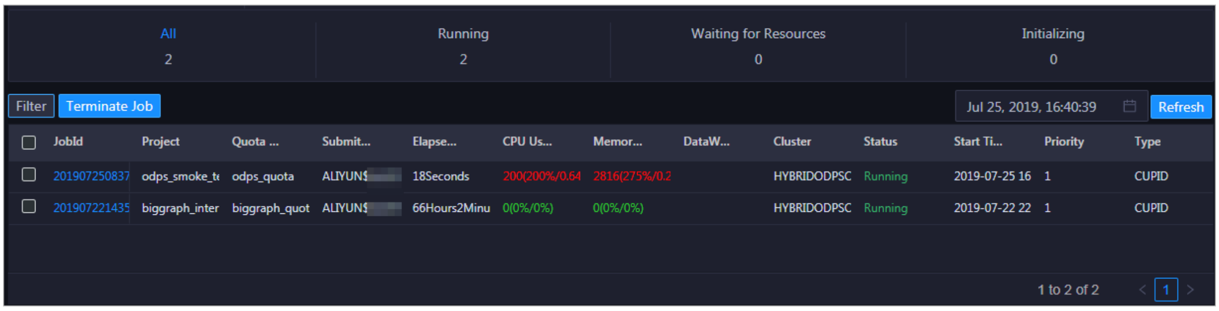
Method of merging small files

ABM uses the MaxCompute SDK to generate merge tasks for merging small files. This method increases merging concurrency to the maximum extent. Currently, you can create merge tasks by cluster or project. You can configure whether to allow merge tasks to run concurrently and specify the start and end time for each merge task.

26.4.2. Job snapshot

This topic describes the job snapshot feature of ABM for MaxCompute.

In this topic, all jobs refer to MaxCompute jobs. When a job is executed, ABM saves detailed job logs. These logs are used to generate a job snapshot. The following figure shows an example of the job snapshot page.



All	Running	Waiting for Resources	Initializing
2	2	0	0

□	JobId	Project	Quota ...	Submit...	Elapse...	CPU Us...	Memor...	DataW...	Cluster	Status	Start Ti...	Priority	Type
□	201907250837	odps_smoke_t	odps_quota	ALIYUN$[redacted]	18Seconds	200(200%/0.64	2816(275%/0.2		HYBRIDODPSC	Running	2019-07-25 16	1	CUPID
□	201907221435	biggraph_inter	biggraph_quot	ALIYUN$[redacted]	66Hours2Minu	0(0%/0%)	0(0%/0%)		HYBRIDODPSC	Running	2019-07-22 22	1	CUPID

The job snapshot feature supports the following functions:

- Displays information about current and historical jobs, including the resource usage and queuing status.
- Supports aggregating jobs from different dimensions, such as the quota group, submitter, and job status. This allows you to clearly understand the status of current jobs.

- Supports generating a detailed Logview page for a single job.
- Supports terminating jobs.

26.4.3. Geo-disaster recovery

This topic describes geo-disaster recovery for big data services.

In geo-disaster recovery for big data services, the primary and secondary clouds are independent of each other. As a result, the account system, O&M management system, and delivery of each cloud are also separate. Geo-disaster recovery for big data services focuses only on offline-side data recovery. Involved services are DataWorks and MaxCompute. The recovery time object (RTO) is 0 and the recovery point object (RPO) is less than or equal to one day.

In geo-disaster recovery for big data services, resources in both the primary and secondary clouds are visible to users. The instances covered by disaster recovery, such as DataWorks projects, are allocated between the primary and secondary data centers in 1:1 mapping. Application-based protection groups are created. In geo-disaster recovery, big data components are deployed in the primary and secondary data centers symmetrically based on business systems.

Business data for offline computing comes from the following sources: Apsara Stack services including ApsaraDB RDS, Object Storage Service (OSS), and Tablestore, and your existing business systems outside Apsara Stack. These data sources use Cloud Data Pipelines (CDPs) that are built in DataWorks to integrate data into MaxCompute for offline big data computing. The computing results are stored in MaxCompute projects or written back to databases such as ApsaraDB RDS databases.

Offline computing in the two data centers utilizes the active-standby mode. This means that data sources outside Apsara Stack are connected with the CDP of only one data center and Apsara Stack services including ApsaraDB RDS, OSS, and Tablestore depend on their own geo-disaster recovery solutions.

To use DataWorks for offline computing on MaxCompute, you must create projects and tasks on DataWorks and store the information in the metadatabase of DataWorks. The data must be synchronized cross-cloud through DataWorks in the primary and secondary data centers because the data is strongly correlated with offline computing. Big data services have a huge amount of business data and the data can be synchronized only once per day. You can configure the specific synchronization time in ABM.

- DataWorks geo-disaster recovery solution

  DataWorks provides interfaces for ABM to export and import metadata, and start and stop task instances. ABM can synchronize metadata across clouds and perform failovers between primary and secondary instances by calling these interfaces. For example, when a switchover is being performed, ABM stops scheduling the offline computing task instances in DataWorks of the secondary data center. Then, ABM extracts the metadata of DataWorks, corrects specific information based on the global information specified by the user in ABM, and then imports the metadata to DataWorks of the secondary data center. Finally, ABM stops the offline computing task instances in the primary data center and starts the task instances in the secondary data center.

- MaxCompute geo-disaster recovery solution

  ABM calls the CopyTask operation to start cross-cloud replication of MaxCompute business data and performs disaster recovery-related management and operations. Cross-cloud replications between the primary and secondary data centers support full and incremental replications because the volume of data for big data components is very large. Partition is the smallest granularity for data replication supported by the CopyTask operation. Therefore, when you design a large table, you must plan your partitions at a smaller granularity such as dates (20181008) or hours (2018100810).

The environment of geo-disaster recovery for big data services depends on the overall Apsara Stack Enterprise geo-disaster recovery configurations, including network connections between the two locations, separation of business and replication flows, DNS forwarding, and GSLB switchover. For more information, see *Technical Whitepaper for Geo-disaster Recovery*.